



AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA INŻYNIERIA MECHANICZNA

ROZPRAWA DOKTORSKA

Automatyzacja oceny stanu materiału z wykorzystaniem
metody SPT

Autor: Maciej Kaliciak

Promotor rozprawy: Prof. dr hab. inż. Tadeusz Uhl
Promotor pomocniczy: Dr inż. Marek Nowak

Praca wykonana: Wydział Inżynierii Mechanicznej i Robotyki

Kraków, 2025

*Szczególne podziękowania przekazuję dla Pana
Prof. dr hab. inż. Tadeusza Uhla za nieocenioną
pomoc, wsparcie merytoryczne oraz motywację
do działania.*

Abstract

This thesis was developed in response to the needs of industry, particularly in the field of mechanical structures intended for long term, uninterrupted operation. There is a growing demand for structural health monitoring of material property degradation over time, and the Small Punch Test (SPT) offers a promising solution. It serves as a bridge between classical destructive testing methods, which provide key mechanical properties of steels, and non-destructive methods, which do not allow such identification.

The current state of knowledge presented in the literature does not offer a complete understanding of the correlation between SPT and the standard uniaxial tensile test (UTT), nor does it address the full automation of material condition assessment. Recent developments in the method tend to explore alternative applications rather than reinforce the position of SPT itself, which was the focus of this study.

The aim of the work was to perform individual and statistical assessments of the degradation of boiler steels 10H2M, 13HMF, and 15HM, followed by the automation of result interpretation. The study addressed the question of whether the SPT method can be reliably used to assess the condition of selected boiler steels, enabling evaluation of their degradation under operational conditions.

As part of the research, SPT and UTT analyses were carried out, and the obtained results were statistically evaluated. Based on these data, material databases were constructed to train machine learning models, and two neural networks were developed for the purpose of automating material condition assessment.

The results confirmed that SPT can be considered an alternative method for estimating yield strength and ultimate tensile strength of boiler steels. Moreover, the method showed sensitivity to local variations in the tested materials. In the automated assessment workflow, a convolutional neural network was selected as the most effective for predicting UTT results based on SPT input, achieving a mean error of 2.9% for yield strength and 0.9% for ultimate tensile strength.

The conducted research advances the state of knowledge regarding the practical application of SPT and introduces a complete, specialized prediction algorithm designed to automate material condition assessment using a novel neural network architecture and real experimental data. The outcomes may serve as a foundation for further developments in the field, where expanding the database to include additional steel grades and classifier models can further improve prediction accuracy.

Streszczenie

Niniejsza praca powstała, aby odpowiedzieć na potrzeby przemysłu, szczególnie w obszarze konstrukcji mechanicznych nastawionych na wieloletnie, nieprzerwane użytkowanie. Rynek wymaga systematycznej kontroli stanu degradacji własności materiałów, a badanie Small Punch Test wychodzi naprzeciw tym potrzebom. Stanowi ono pomost pomiędzy badaniami klasycznymi, niszczącymi, które dostarczają informacji o głównych własnościach badanych stali, a badaniami nieniszczącymi, które nie identyfikują tych parametrów.

Obecny stan wiedzy opisany w literaturze nie daje pełnego poglądu na korelację pomiędzy SPT a klasyczną próbą jednoosiowego rozciągania (UTT), a także nie rozwija zagadnień pełnej automatyzacji oceny stanu materiału. Trendy rozwoju metody przechodzą w jej szersze i alternatywne zastosowania, a nie umacniają pozycji samego badania, co zostało zrealizowane w ramach niniejszej pracy.

Jako cel pracy postawiono jednostkową oraz statystyczną ocenę degradacji stali kotłowych 10H2M, 13HMF oraz 15HM, a następnie automatyzację interpretacji wyników tej oceny. Odpowiedziano na pytanie, czy stosując metodę SPT, można dokonać wiarygodnej analizy stanu materiału wybranych stali kotłowych, pozwalającej na ocenę ich degradacji w warunkach eksploatacyjnych.

W ramach badań przeprowadzono analizy SPT oraz UTT, a następnie statystycznie przeanalizowano otrzymane wyniki. Na ich podstawie zbudowano bazy danych, które posłużyły do treningu modeli opartych na uczeniu maszynowym oraz opracowano dwie sieci neuronowe, przygotowane do zadania automatyzacji oceny stanu materiału.

Otrzymane wyniki pozwoliły na scharakteryzowanie SPT jako alternatywnej metody wyznaczania granicy plastyczności i granicy wytrzymałości stali kotłowych. Wskazały dodatkowo na jej czułość na lokalne zmiany w badanych materiałach. W dalszym porównaniu wyników z procesu automatyzacji oceny stanu materiału wybrano konwolucyjną sieć neuronową jako najlepszą do predykcji wyników UTT na podstawie SPT, z błędem średnim wynoszącym 2.9% dla granicy plastyczności oraz 0.9% dla granicy wytrzymałości.

Przeprowadzane badania poszerzają stan wiedzy o praktycznym zastosowaniu SPT oraz przedstawiają kompletny, wyspecjalizowany algorytm predykcji, służący do automatyzacji oceny stanu materiału na podstawie nowatorskiej architektury sieci neuronowych i rzeczywistych wyników eksperymentalnych. Opisane prace mogą stanowić podstawę do dalszych badań w dziedzinie oceny stanu materiału, a dalsze rozszerzanie bazy danych o nowe gatunki stali i dodatkowe klasyfikatory może zwiększyć precyzję predykcji.

Spis treści

Spis rysunków	xi
Spis tabel	xv
1 Wstęp	1
1.1 Cel i teza pracy	1
1.2 Struktura pracy	2
2 Obecny stan badań nad rozwojem i zastosowaniem metody SPT	3
2.1 Historia Small Punch Test (SPT)	3
2.2 Konwencjonalne metody referencyjne	7
2.2.1 Statyczna próba rozciągania	7
2.2.2 Badanie udarności	9
2.2.3 Statyczna próba pełzania	11
2.3 Nieniszczące metody diagnostyki stanu materiału	12
2.3.1 Magnetic Barkhausen Noise (MBN)	12
2.3.2 Badanie mikrotwardości	13
2.3.3 Metody ultradźwiękowe	14
2.3.4 Metoda prądów wirowych	15
2.4 Small Punch Test (SPT)	16
2.5 Small Punch Creep Test (SPCT)	21
2.6 Inne wybrane parametry materiałowe	23
2.6.1 DBTT	23
2.6.2 Wydłużenie	25
2.6.3 Moduł sprężystości	25
2.7 Zastosowanie metody SPT dla różnych materiałów	26
2.8 Sieci neuronowe (NN) i uczenie maszynowe	30
2.9 Standardowa metodologia analizy wyników testów SPT	33
2.10 Aktualne trendy w rozwoju metody SPT	39
2.10.1 SPT na zaawansowanych materiałach	39
2.10.2 Alternatywne zminiaturyzowane metody testowe	40
2.10.3 SPCT	40
2.10.4 SPT w środowiskach agresywnych	41

2.10.5	Numeryczne i empiryczne podejścia do badania SPT	42
2.11	Podsumowanie przeglądu literatury	42
3	Metodologia badań	45
3.1	Materiały wybrane do badań	45
3.2	Przygotowanie próbek	50
3.3	Procedura eksperymentalna	55
3.3.1	SPT	55
3.3.2	SPCT	58
3.3.3	Klasyczne badanie UTT	61
3.3.4	Analiza mikroskopowa	61
3.4	Metody analizy wyników eksperymentalnych	62
3.4.1	Interpretacja danych	62
3.4.2	Statystyczna analiza wyników badań porównawczych	63
3.5	Analiza komputerowa i automatyzacja oceny stanu materiału	67
3.5.1	Przetwarzanie danych eksperymentalnych	70
3.5.2	Budowa baz danych z wynikami eksperymentów	72
3.5.3	Sieć neuronowa	75
3.5.4	Automatyzacja oceny stanu materiału	81
3.5.5	Automatyzacja analizy statystycznej	87
4	Wyniki badań	89
4.1	Badania eksperymentalne i porównanie metod dla badanych materiałów	89
4.1.1	Stal 10H2M	89
4.1.2	Stal 13HMF	103
4.1.3	Stal 15HM	109
4.2	Modelowanie danych z użyciem Sieci Neuronowej i analiza wyników predykcji	114
4.2.1	Stal 10H2M	114
4.2.2	Stal 13HMF	118
4.2.3	Stal 15HM	121
4.2.4	Podsumowanie wyników predykcji	124
5	Podsumowanie i wnioski	127
5.1	Dyskusja i interpretacja wyników w kontekście celu pracy	127
5.2	Ograniczenia, potencjalne usprawnienia i rekomendacje	128
A	Kod źródłowy	131
	Bibliografia	193

Spis rysunków

2.1	Schematyczne przedstawienie zestawu testowego i otrzymywanego w badaniu SPT wykresu. [78]	4
2.2	Schematyczne przedstawienie cierno-krawędziowej czaszy tnącej do pobierania próbek materiału. [18]	5
2.3	Kształt próbek o przekroju okrągłym: (a) przed testem; (b) po teście. [50]	8
2.4	Inżynierska krzywa rozciągania w układzie naprężenie - odkształcenie.	9
2.5	Zakres plastyczny na inżynierskim wykresie naprężenie-odkształcenie	10
2.6	Standardowe wymiary próbki do testu udarowości [32].	11
2.7	Uproszczony schemat wykorzystania UT do wykrywania pęknięcia w materiale [97]	14
2.8	Graficzne przedstawienie metodologii analizy krzywej obciążenie-pocienienie próbki. [55]	18
2.9	Istniejące metody wyznaczania przejścia materiału w stan plastyczny w badaniu SPT. [55]	19
2.10	Symulacje MES cykli obciążania i odciążania próbki. [27]	26
2.11	Schematyczna ilustracja a) strategii skaningowej z rotacją 67° pomiędzy warstwami; b) orientacji i wymiarów próbki SPCT pobranej z bloku materiału GH4169. [83]	30
2.12	Identyfikacja parametrów materiałowych z użyciem sieci neuronowych. [78]	31
2.13	Po lewej: struktura sieci neuronowej. Po prawej: każda sieć jest nauczana na danym przemieszczeniu. [78]	32
2.14	Wyznaczanie wartości F_e oraz u_e metodą bi-linearną [18].	34
2.15	Korelacja pomiędzy wartością F_m uzyskaną w próbie SPT oraz znanymi wartościami granicy wytrzymałości R_m badanych materiałów. [9]	35
2.16	Przykładowa krzywa wyznaczająca wartość T_{SP} z użyciem znormalizowanej energii dla stali P92. Wynikowa wartość $T_{SP} = -116^\circ\text{C}$ zaznaczona na wykresie.	37
2.17	Zależność wstępnej wielkości nacięcia od przemieszczenia stempla w próbie SPT. [5]	38
2.18	Przykładowe zestawienie wartości naprężenia pełzania dla stali P92 w teście SPCT z wartościami z klasycznej próby pełzania interpolowanymi modelem Wilshire'a [43, 152]	39
3.1	Schemat blokowy do zagadnienia automatyzacji oceny stanu materiału	46
3.2	Schemat blokowy przygotowania próbek SPT	51
3.3	Proces pobierania próbki	51
3.4	Jednostka sterująca urządzeniu tnącemu.	52
3.5	Czasza tnąca	53

3.6	Próbka materiału w kolejnych etapach przygotowania: a) łożka pobrana z materiału bazowego; b) płaski plaster wycięty z łożki; c) pozostałość materiału po wybiciu próbek SPT na prasie.	53
3.7	Szlifierka laboratoryjna	54
3.8	Narzędzia do szlifowania próbek SPT: zestaw pierścieni, kamień szlifierski, dociski.	54
3.9	Schemat próbki do badania UTT	55
3.10	Maszyna wytrzymałościowa TTS 190	56
3.11	Próbka na matrycy, wyjęta z maszyny testowej SPT	57
3.12	Komora kriogeniczna w czasie badania SPT w niskich temperaturach.	58
3.13	Maszyna SPC 800	59
3.14	Kartridż na próbkę do badania SPCT	60
3.15	Trzpień testowy maszyny TTS 190	60
3.16	Przykładowa krzywa SPT	62
3.17	Schemat blokowy przygotowania wyników SPT i UTT do implementacji w bazie danych	70
3.18	Wycinek struktury bazy danych	74
3.19	Krzywe uczenia i walidacji modelu MLP	78
3.20	Wykres uczenia i walidacji na modelu 1D CNN	81
3.21	Schemat blokowy pracy sieci neuronowej	83
3.22	Przykładowy wykres wygenerowany przez sieć neuronową	85
4.1	Materiał badawczy J1 - 10H2M w stanie przepracowanym.	90
4.2	Mapy ciepła, obrazujące zależność grubości próbki testowej z: a) Granicą plastyczności; b) Granicą wytrzymałości.	91
4.3	Histogramy wyników D1 - Granica plastyczności	92
4.4	Histogramy wyników D1 - Granica wytrzymałości	93
4.5	Histogramy wyników J1 - Granica plastyczności	93
4.6	Histogramy wyników J1 - Granica wytrzymałości	94
4.7	Wykresy prawdopodobieństwa rozkładu dla stali 10H2M - Granica plastyczności	95
4.8	Wykresy prawdopodobieństwa rozkładu dla stali 10H2M - Granica wytrzymałości	96
4.9	Dendrogram z analizy kłastrów dla materiału 10H2M	96
4.10	Zdjęcia SEM mikrostruktury próbki J1 - strona wewnętrzna	99
4.11	Zdjęcia SEM mikrostruktury próbki J1 - strona zewnętrzna	99
4.12	Zdjęcia SEM mikrostruktury próbki J1 - przekrój poprzeczny	99
4.13	Krzywa DBTT dla materiału 10H2M w wariancie D1	100
4.14	Krzywa DBTT dla materiału 10H2M w wariancie J1	101
4.15	Próbki J1 DBTT: -160C po lewej; -165C po prawej.	101
4.16	Próbki D1 DBTT: -175C po lewej; -170C po prawej.	101
4.17	Wykres zależności minimalnej prędkości pełzania od czasu do pęknięcia próbki w teście SPCT	103
4.18	Histogramy wyników M3 - Granica plastyczności	104

4.19	Histogramy wyników M3 - Granica wytrzymałości	105
4.20	Histogramy wyników R1 - Granica plastyczności	105
4.21	Histogramy wyników R1 - Granica wytrzymałości	106
4.22	Wykresy prawdopodobieństwa rozkładu dla stali 13HMF - Granica plastyczności	106
4.23	Wykresy prawdopodobieństwa rozkładu dla stali 13HMF - Granica wytrzymałości	107
4.24	Dendrogram z analizy klastrow dla materiału 13HMF	108
4.25	Histogramy wyników H1 - Granica plastyczności	110
4.26	Histogramy wyników H1 - Granica wytrzymałości	110
4.27	Histogramy wyników T1 - Granica plastyczności	111
4.28	Histogramy wyników T1 - Granica wytrzymałości	111
4.29	Wykresy prawdopodobieństwa rozkładu dla stali 15HM - Granica plastyczności	112
4.30	Wykresy prawdopodobieństwa rozkładu dla stali 15HM - Granica wytrzymałości	112
4.31	Dendrogram z analizy klastrow dla materiału 15HM	113
4.32	Wykresy prawdopodobieństwa dla danych z predykcji granicy plastyczności stali 10H2M . .	117
4.33	Wykresy prawdopodobieństwa dla danych z predykcji granicy wytrzymałości stali 10H2M .	117
4.34	Przykładowe zestawienie wyników predykcji z rzeczywistą krzywą UTT dla materiału J1 . .	118
4.35	Zbliżenie na zakres sprężysty wykresów z predykcji NN	119
4.36	Wykresy prawdopodobieństwa dla danych z predykcji granicy plastyczności stali 13HMF . .	121
4.37	Wykresy prawdopodobieństwa dla danych z predykcji granicy wytrzymałości stali 13HMF .	122
4.38	Wykresy prawdopodobieństwa dla danych z predykcji granicy plastyczności stali 15HM . .	124
4.39	Wykresy prawdopodobieństwa dla danych z predykcji granicy wytrzymałości stali 15HM . .	125

Spis tabel

2.1	Lista korelacji wyników SPT z granicą plastyczności materiałów. [14]	6
2.2	Lista korelacji wyników SPT z granicą wytrzymałości materiałów. [14]	6
2.3	Zależność współczynnika β_{R_m} od geometrii próbki i zestawu testowego. [18]	35
2.4	Proponowane wartości współczynnika $\beta_{R_{p0,2}}$ dla standardowej geometrii zestawu testowego i danych parametrów próbki. [18]	36
3.1	Zbiór materiałów badanych w laboratorium SPT w Centrum Energetyki AGH wraz z oznaczeniem kodowym oraz czasem eksploatacji danej próbki.	48
3.2	Procentowy skład chemiczny badanych materiałów. [144, 138, 92, 158]	48
3.3	Właściwości materiałów badanych w laboratorium SPT w stanie nowym normalizowanym. [144, 138, 92, 158]	49
3.4	Zestawienie liczby próbek SPT i UTT zaimplementowanych do bazy danych.	73
3.5	Metryki wydajności modelu MLP.	78
3.6	Tabela podziału na zbiory danych do uczenia, walidacji i testów.	80
3.7	Metryki zgodności modelu CNN dla danego materiału.	82
3.8	Wyniki testów klasyfikatora materiału	87
4.1	Wartości wyników badań metodami SPT i UTT dla materiału 10H2M.	91
4.2	Wyniki analizy klastrów dla całej populacji próbek 10H2M	95
4.3	Centroidy dla wszystkich próbek SPT materiału 10H2M	97
4.4	Centroidy dla 4 grup próbek SPT materiału 10H2M	97
4.5	Centroidy dla wszystkich próbek UTT materiału 10H2M	98
4.6	Wyniki SPCT dla stali 10H2M	102
4.7	Średnie wartości wyników badań metodami SPT i UTT dla materiału 13HMF.	104
4.8	Centroidy klastrów materiału 13HMF.	108
4.9	Średnie wartości wyników badań metodami SPT i UTT dla materiału 15HM.	109
4.10	Centroidy klastrów materiału 15HM.	113
4.11	Tabela porównawcza dla wyników predykcji dla materiału 10H2M	115
4.12	Wyniki walidacji próbek ze zbioru testowego 10H2M	115
4.13	Podsumowanie testów dopasowania normalnego dla stali 10H2M	116
4.14	Tabela porównawcza dla wyników predykcji dla materiału 13HMF	120
4.15	Wyniki walidacji próbek ze zbioru testowego 13HMF	120
4.16	Podsumowanie testów dopasowania normalnego dla stali 13HMF	120

4.17	Tabela porównawcza dla wyników predykcji dla materiału 15HM	122
4.18	Wyniki walidacji próbek ze zbioru testowego 15HM	123
4.19	Podsumowanie testów dopasowania normalnego dla stali 15HM	123
4.20	Odchylenie średniej wyników R_e ze wszystkich źródeł względem wyników UTT	125
4.21	Odchylenie średniej wyników R_m ze wszystkich źródeł względem wyników UTT	126

Rozdział 1

Wstęp

1.1 Cel i teza pracy

Współczesne tendencje rozwoju techniki oraz przemysłu, szczególnie w obszarach energetyki i konstrukcji mechanicznych, wymagają niezawodnych metod oceny stanu materiałów stosowanych w eksploatacji. Konwencjonalne metody badań, choć dokładne, często nie pozwalają na szybkie, ekonomiczne i minimalnie inwazyjne ocenienie degradacji materiału podczas użytkowania urządzeń czy instalacji przemysłowych. W odpowiedzi na te potrzeby rozwijana jest metoda Small Punch Test (SPT), pozwalająca na oszacowanie właściwości mechanicznych oraz stopnia degradacji przy minimalnym wpływie na badany element.

Cel niniejszej rozprawy to opracowanie metodyki oceny stanu degradacji stali na podstawie wyników testów SPT oraz zaproponowanie zautomatyzowanego procesu interpretacji tych wyników. W ramach realizacji celu głównego uwzględniono następujące cele cząstkowe:

- Ocena degradacji – określenie wpływu długotrwałej eksploatacji oraz oddziaływań środowiskowych na właściwości mechaniczne materiałów, z wyróżnieniem stali 10H2M, 13HMF oraz 15HM.
- Automatyzacja interpretacji wyników – opracowanie narzędzi w postaci modeli numerycznych i algorytmów, umożliwiających szybkie, powtarzalne i niezależne od operatora przetwarzanie wyników badań metodą SPT.

Wysoka wiarygodność metod oceny degradacji materiałów konstrukcyjnych jest kluczowym czynnikiem gwarantującym bezpieczeństwo i niezawodność infrastruktury technicznej. Metoda SPT, dzięki swojej kompaktowej formie oraz minimalnym wymaganiom materiałowym, stanowi atrakcyjne rozwiązanie, pozwalające na diagnostykę stanu technicznego elementów bez konieczności ich wycofywania z eksploatacji. Dlatego zasadne jest sprawdzenie możliwości praktycznego wykorzystania tej metody dla materiałów szczególnie eksploatowanych w wymagających warunkach, takich jak stal kotłowa.

Teza badawcza: Stosując metodę SPT, można dokonać wiarygodnej analizy stanu materiału wybranych stali kotłowych, pozwalającej na ocenę ich degradacji w warunkach eksploatacyjnych. Realizacja tego zagadnienia obejmuje zarówno weryfikację korelacji pomiędzy wynikami SPT a właściwościami mechanicznymi materiału, uzyskanymi z klasycznych prób mechanicznych, jak również opracowanie zintegrowa-

nego systemu interpretacji wyników badań do oceny aktualnego stanu materiału, co może przyczynić się w przyszłości do przewidywania zmian własności materiałów w porównaniu z wymaganiami.

1.2 Struktura pracy

Struktura rozprawy została zbudowana wokół trzech głównych filarów badawczych, które wzajemnie się uzupełniają i razem tworzą spójne rozwinięcie postawionej tezy.

Pierwszy filar stanowi analiza literaturowa dotycząca metody SPT oraz powiązanych zagadnień, takich jak metody odniesienia, właściwości mechaniczne materiałów, degradacja stali, a także aktualne podejścia do interpretacji wyników SPT. W tej części zawarto również przegląd zastosowań sztucznej inteligencji, w szczególności sieci neuronowych, w analizie danych materiałowych.

Drugi obszar obejmuje analizę wyników eksperymentalnych uzyskanych w wyniku przeprowadzenia testów SPT i SPCT dla próbek stali 10H2M. Uwzględnia on opis zastosowanej aparatury, procedury badawczej oraz szczegółowe opracowanie danych uzyskanych w ramach badań fizycznych. Celem tej części jest wyznaczenie charakterystycznych parametrów mechanicznych oraz identyfikacja ich zmian w zależności od stopnia degradacji materiału.

Trzecim i kluczowym elementem rozprawy jest część poświęcona budowie i walidacji modelu sieci neuronowej, którego zadaniem jest automatyczna interpretacja wyników SPT i predykcja parametrów materiałowych. W tej części przedstawiono etapy tworzenia modelu, dobór architektury, sposób trenowania oraz ocenę skuteczności rozwiązania. Integracja wyników eksperymentalnych z wynikami predykcji komputerowej stanowi istotny krok w kierunku pełnej automatyzacji procesu oceny degradacji.

Podział pracy na powyższe segmenty odzwierciedla interdyscyplinarny charakter rozprawy, łącząc wiedzę z zakresu inżynierii materiałowej, metod eksperymentalnych w mechanice oraz nowoczesnych metod analizy danych.

Rozdział 2

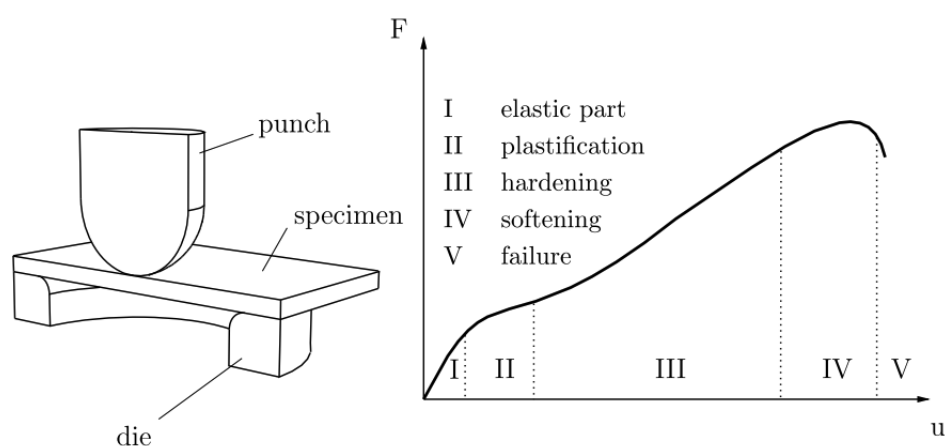
Obecny stan badań nad rozwojem i zastosowaniem metody SPT

Techniczne standardy związane z rozwojem przemysłu stawiają wysokie wymagania co do badań prowadzonych na eksploatowanych materiałach. Każde wstrzymanie produkcji bądź wyłączenie komponentu z użytku w celu pobrania próbki materiału stanowi straty finansowe dla zlecającej firmy. Pobór próbek do konwencjonalnych badań niezbędnych do określenia stanu materiału często wymaga wstrzymania eksploatacji i wymiany całych segmentów komponentów. Wymusiło to rozwój niekonwencjonalnych metod badań materiałów, do których należy metodologia SPT. Badania SPT, ze względu na niewielką ilość materiału potrzebnego do wykonania próbki, należą do grupy badań nieniszczących. W ostatnich latach powstaje coraz więcej publikacji traktujących o porównaniu otrzymanych wyników z wartościami uzyskiwanymi równocześnie z pomocą konwencjonalnych metod badawczych oraz porównaniu wyników metod symulacji komputerowych. Na tym polu wciąż pozostają obszary nieprzeanalizowane, wymagające doprecyzowania posiadanych informacji w celu walidacji metod SPT, aby umożliwić ich zastosowanie w szerokim zakresie przemysłowym. Mimo to, komercjalizacja tych badań jest zagadnieniem postępującym, ponieważ coraz większe zainteresowanie widocznymi korzyściami ze strony przemysłu wymusza na laboratoriach materiałowych zakup specjalistycznego oprzyrządowania wymaganego do przeprowadzenia wszystkich badań, co tym samym przyspiesza ich rozwój. W przyszłości, efektem takiej dynamiki współpracy, będzie szeroki dostęp do metody SPT, a z czasem możliwe wypchnięcie konwencjonalnych metod badawczych z pozycji dominującej na rynku [62].

2.1 Historia Small Punch Test (SPT)

Metoda SPT (2.1) jest to rodzaj badania mający swoje początki już w latach osiemdziesiątych. Powstała ona z konieczności prowadzenia analiz właściwości na napromieniowanych materiałach [82]. Poszukiwania sposobu na uzyskanie wyników odpowiadającym standardowym próbom jednoosiowego rozciągania, które uznaje się za największe obciążenie możliwe do przeniesienia przez materiał, przyczyniły się do miniaturyzacji próbek sięgających nawet 3 mm średnicy oraz 0.25 mm grubości [86]. Rozmiar ten zbliżony jest do próbek stosowanych w transmisyjnej mikroskopii elektronowej. Metodyka ta została jako pierw-

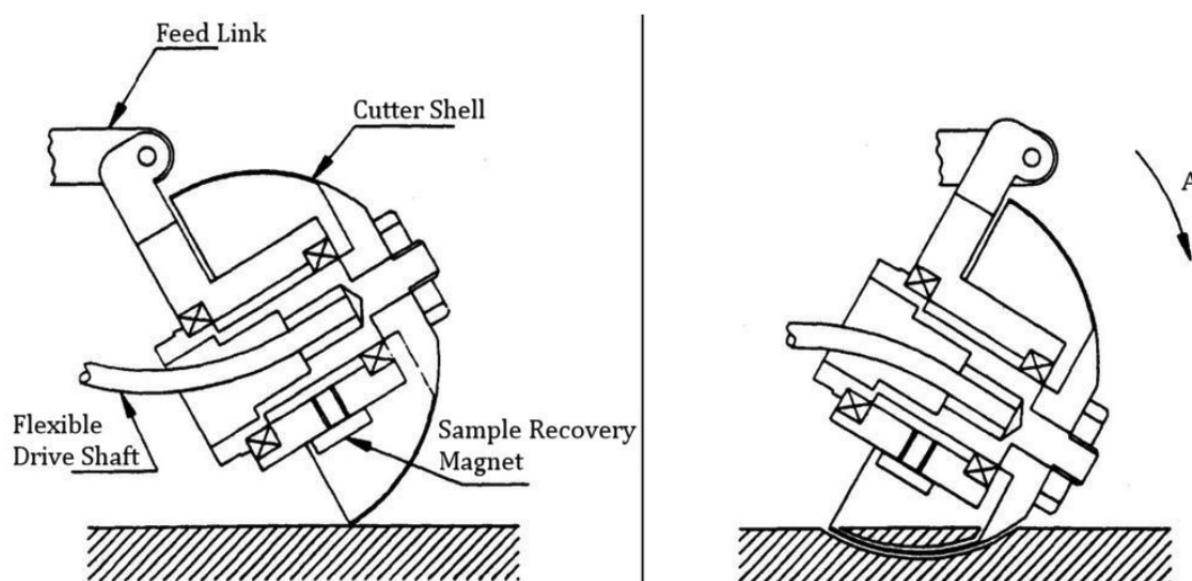
sza opatentowana w Stanach Zjednoczonych w 1983 roku jako Miniature Disk Bend Test (MDBT) i na tej podstawie prowadzono dalszy rozwój badań materiałowych z wykorzystaniem miniaturowych próbek. Z początku wykorzystywano próbki o przekroju kwadratu 10x10x0.5 mm. Próbki te wykorzystywane były w celu określenia parametrów materiałowych takich jak temperatura przejścia w stan kruchy stali napromieniowanych neutronami w elektrowniach jądrowych. Próbka w trakcie testu była nie tylko silnie rozciągana, ale również częściowo zrywana aż do makroskopowego pęknięcia. Obciążenie zadawane było przez stalową kulkę o średnicy 2.4 mm z prędkością przemieszczenia 0.02 mms^{-1} . Wyniki otrzymane tą metodą można było skorelować ze standardową próbą Charpy'ego z wykorzystaniem stanu naprężenia, jak i prędkości odkształcenia. Pierwszy raz również zwrócono uwagę na pracę stempla jako energię procesu do określenia momentu przejścia w stan kruchy.



Rysunek 2.1: Schematyczne przedstawienie zestawu testowego i otrzymywanego w badaniu SPT wykresu. [78]

Europejska norma EN 10371:2021 [18] opisuje metody uznane za odpowiednie do charakteryzacji poszczególnych parametrów materiałowych z wykorzystaniem badań z grupy SPT. Opisuje ona również schematy przeprowadzania prób SPC oraz SPCT i analizę otrzymanych wyników. Mimo iż na przestrzeni lat zrealizowano znaczną liczbę badań związanych z badaniem porównania z klasycznymi metodami, a nawet konwersją wyników otrzymywanych w badaniach typu SPT i klasycznych, to jedynie kilka z nich zostało zatwierdzonych w normie. Metody te wciąż jednak wymagają udoskonalania, gdyż ich dokładność potwierdzona jest konkretnymi badaniami w zakresach, które w warunkach laboratoryjnych często są przekraczane. Wychodzenie poza przewidziane zakresy zastosowania pozwala zbadać skalę odchyień do punktu wyjścia, jakim są wartości znajdujące się w normie. Aby metoda mogła zostać uznana za zdatną do zastosowania w warunkach przemysłowych, musi spełnić rygorystyczne wymagania stawiane metodom stosowanym dla materiałów w agresywnych środowiskach. Do takich celów opracowano metodę SPT i jest to niezwykle istotne, aby jej wyniki były zbieżne z klasycznymi metodami. W przypadku zastosowań dla rurociągów pracujących pod dużym stałym i zmiennym obciążeniem, w materiale następują nieodwracalne zmiany, niemożliwe do określenia z pomocą standardowych metod analizy materiałowej. Dla takich materiałów stosuje się niekonwencjonalne metody nieniszczące, zapewniające brak ingerencji lub znikomą ingerencję w mocno już wyczerpaną strukturę danego obiektu. W przypadku badań SPT wymagane jest pobranie niewielkiej

objętości materiału w postaci wyoblanego krążka o geometrii ścinka kuli. Taki kształt materiału uzyskany jest poprzez zastosowanie tzw. Scoopera (Rysunek 2.2), czyli robota wykorzystującego wirującą czaszę do wycięcia z powierzchni danego obiektu fragmentu materiału, nie ingerując w jego integralność strukturalną. Dzięki sferycznej geometrii obszar pozostały po usunięciu materiału do badań nie jest zagrożony inicjacją ani propagacją już istniejących pęknięć oraz nie pozostawia ostrych krawędzi podatnych na uszkodzenia. Niewielka głębokość (do 3.5 mm) wcięcia w głąb badanego obiektu nie wpływa na osłabienie ogólnej wytrzymałości rurociągów o standardowej grubości ścianek. To pozwala na wpisywanie badań SPT do grupy badań nieniszczących, mimo rzeczywistemu pobieraniu pewnej objętości materiału do badań. Z jednej tak pobranej próbki powstać może około 7-8 rzeczywistych próbek testowych. Taka liczba zapewnia możliwość wielokrotnego przeprowadzenia próby w danych warunkach lub też zapewnia zapas próbek do określenia własności mechanicznych, odporności na pełzanie oraz określania temperatury przejścia materiału w stan kruchy. Na potrzeby badań laboratoryjnych wycina się znacznie więcej fragmentów materiału. Jeśli jednak materiał bazowy nie jest ograniczony, jak w przypadku zleceń przemysłowych, możliwe jest pozyskiwanie zapasu próbek testowych z pomocą standardowych metod oraz wycinanie ich z posiadanych już fragmentów materiału po klasycznych testach, takich jak próba Charpy'ego. Przełamane i potencjalnie przeznaczone do utylizacji fragmenty materiału stanowić mogą doskonałe źródło zapasu materiału badawczego.



Rysunek 2.2: Schematyczne przedstawienie cierno-krawędziowej czaszy tnącej do pobierania próbek materiału. [18]

Badania prowadzone nad metodami SPT można podzielić na okres przed stanowieniem standardu dla badania w Europie oraz po tym czasie. Za początek powstawania metody SPT jedną z bardziej przełomowych publikacji, do której odwoływano się przez wiele lat [29] jest artykuł Mao i Takahashi'ego [86] z roku 1987. W pracy tej potwierdzono konieczność stosowania coraz mniejszych miniaturowych próbek do badań, ze szczególnym naciskiem na dyski o średnicy 3 mm i grubości 0.25 mm, pozwalające na bezpieczną analizę materiałową napromieniowanych obiektów. Przeprowadzono badania w kierunku określenia procesu deformacji, równoważnego odkształcenia pęknięcia oraz wpływu jego wielkości na pomiary przy dwuosiowym

stanie naprężenia [105]. Analizowano również relację pomiędzy odkształceniem pęknięcia a odpornością na pęknięcie i opracowywano metodę wyznaczenia granicy plastyczności i wytrzymałości badanych materiałów. Już wtedy opracowano standardy wymiarów zestawu testowego, wykorzystywane do dziś. Badania przeprowadzono dla stali stopowych: SUS316, PCA oraz HT-60. Dla wspomnianych materiałów potwierdzono liniową zależność wartości wytrzymałości na pęknięcie JIC i odkształcenia pęknięcia $\bar{\epsilon}_{qf}$ przy dwuosowym stanie naprężenia.

Doskonałym dodatkiem podsumowującym rozwój metody SPT i stosowanych na przestrzeni lat zestawienia wyników tejże metody z wartościami własności mechanicznych materiałów otrzymywanych w próbie jednoosiowego rozciągania jest zestawienie przedstawione przez Arunkumar Subbaiah w pracy [14]. Zestawienie to, Tablica 2.1 i 2.2 wskazuje autorów danej metody oraz stosowane w niej wzory do wyznaczania parametrów materiału.

Reference	Yield strength relation	Method
Mao and Takahashi	$\sigma_y = 360P_y/t_0^2$	Two tangent method
Xu and Zhao	$\sigma_y = 0.477P_y/t_0^2$	
Cheon and Kim	$\sigma_y = P_y/0.39$ for 0.5 mm thick specimen, $\sigma_y = P_y/0.083$ for 0.25 mm thick specimen	Offset method, $t_0/100$
Fleury and Ha	$\sigma_y = 4.512 P_y/4\pi r_a t_0$, r_a is the radius of contact area between punch tip and specimen	
Ruan et al.	$\sigma_y = (413 \pm 68)P_y/t_0^2 + (149 \pm 108)$, $80 \leq T \leq 293$ K	
Lee and Kim	$\sigma_y = 0.231P_y/t_0^2 + 68.1$	Offset method, $t_0/20$ for circular and $t_0/15$ for square
Eskner and Sandstrom	Plate bending theory	Offset method, 0.5 μm offset from the tangent of the initial linear slope
Finarelli et al.	$\sigma_y = \alpha P_y/t_0^2$, $\alpha = 0.36-0.41$	Two tangent method
Contreras et al.	$\sigma_y = 0.268P_y/t_0^2 + 207$, $-60 \leq T \leq 50$ °C	Offset method, $t/10$
Isselin and Shoji	$\sigma_y = \sqrt{\beta E_e}$, $\beta = 378, 115$, E_e is elastic energy ($R^2 = 0.95$)	Energy method
Garcia et al.	$\sigma_y = \gamma P_y/t_0^2$, $\gamma = 0.346$ for the tested materials with $R^2 = 0.922$	Offset method, $t/10$

Tabela 2.1: Lista korelacji wyników SPT z granicą plastyczności materiałów. [14]

Reference	Ultimate tensile strength relation
Mao and Takahashi	$\sigma_{uts} = 130(P_{\max}/t_0^2) - 320$, $P_{\max}$ or F_m is the maximum load.
Xu and Zhao	$\sigma_{uts} = n^n K$, K is strength coefficient, n is strain hardening exponent
Ruan et al.	$\sigma_{uts} = (77 \pm 15)P_{\max}/t_0^2 + (218 \pm 101)$, $133 \leq T \leq 293$ K
Lee and Kim	$\sigma_{uts} = 0.071P_{\max}/t_0^2 + 60.03$
Contreras et al.	$\sigma_{uts} = 0.051P_{\max}/t_0^2 + 269$, $-60 \leq T \leq 50$ °C
Garcia et al.	$\sigma_{uts} = 0.277P_{\max}/(d_{\max} t_0)$, $R^2 = 0.94$, $d_{\max}$ displacement corresponding to $P_{\max}$
Hong et al.	$\sigma_{uts} = 2.15 P_{\max} + 228$
Altstadt et al.	$R_m = \beta_m F_t/t_0^2$, $\beta_m = 0.183$ for the tested materials with $R^2 = 0.870$,

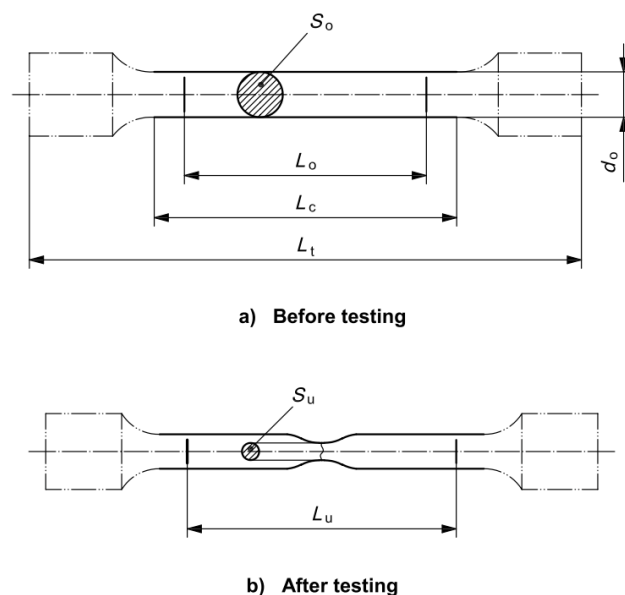
Tabela 2.2: Lista korelacji wyników SPT z granicą wytrzymałości materiałów. [14]

2.2 Konwencjonalne metody referencyjne

Aby wprowadzić pionierską metodę badawczą na rynek komercyjnych badań materiałowych, metoda ta musi być zdolna konkurować z metodami powszechnie zakorzenionymi w przemyśle i w nauce. Aby nowa metoda badania została skutecznie wprowadzona, należy znaleźć dla niej niszę, w której badania konwencjonalne są niewystarczające lub trudne do wykonania. Dla badania SPT taką niszą są zastosowania w przemyśle jądrowym, w którym materiały napromieniowane stanowią zagrożenie zdrowia dla personelu badawczego, oraz w konwencjonalnym przemyśle energetycznym, gdzie specyfika branży nie pozwala na wyłączanie badanych komponentów z użytku. W branżach tych kluczowe są własności wytrzymałościowe eksploatowanych materiałów, które standardowo sprawdza się z wykorzystaniem niszczących metod badawczych. Próbkę do wszystkich opisanych testów są znaczącej wielkości, co wymaga niejednokrotnie zaprzestania użytkowania danej konstrukcji celem wykonania badania, a jej powrót do dalszej eksploatacji jest ograniczony ze względu na akcje naprawcze konieczne do uzupełnienia braku materiału lub też całkowite jego wycofanie z użytkowania.

2.2.1 Statyczna próba rozciągania

W przemyśle, jak i w nauce o badaniach materiałowych, podstawową metodą określania dwóch głównych dla konstruktorów parametrów materiałowych jest statyczna próbka jednoosiowego rozciągania (z ang. Uniaxial Tensile Test - UTT). Pomiędzy innymi parametrami, pozwala ona określić, w sposób standaryzowany, granicę wytrzymałości materiału oraz jego granice plastyczności - umowną górną i dolną w zależności od badanego materiału. Dzięki tej metodzie urzędy dozoru technicznego na całym świecie decydują o dopuszczaniu konstrukcji mechanicznych do użytku, a badacze w dziedzinie inżynierii materiałowej oceniają skuteczność działań pod kątem usprawnienia materiałów metodami przeróbki plastycznej lub cieplnej. Aby wykonać próbę jednoosiowego rozciągania, laboratorium badawcze winno być wyposażone w maszynę wytrzymałościową z tensometrem oraz ekstensometrem w celu zapisu przebiegu siły w funkcji czasu. Maszyny te zaliczane są do grupy uniwersalnych maszyn, a do wykonania testu dodatkowym wymogiem jest odpowiednie zamocowanie próbki w obszarze roboczym z użyciem specjalnie przygotowanych do geometrii próbki szczęk zaciskających. W ramach usprawnienia maszyny testowej dodaje się komponenty do analizy DIC (z ang. Digital Image Correlation) tj. cyfrowej korelacji obrazu [115] co pozwala na usprawnienie metody poprzez rejestrowanie odkształceń materiału w czasie rzeczywistym oparte o pomiar przesunięcia punktów na obrazie w trakcie wykonywania próby rozciągania. Podejście to pozwala precyzyjniej analizować procesy zachodzące w próbkach podczas testu i poprawia dokładność otrzymywanych wyników. Próbkę do testu przygotowuje się według standaryzowanych norm przemysłowych [50] najczęściej w kształcie walców lub płaskich profili (Rysunek 2.3) przygotowanych do testu. Przewężony obszar badany próbki zapewnia, że pęknięcie nastąpi w badanym i obserwowanym obszarze. Aby właściwie określić wydłużenie po teście, próbki oznacza się na powierzchni widocznymi liniami na dystansie L_0 , jednak nie tworzy się nacięć, aby nie generować przedwczesnych pęknięć. Odległość między liniami jest następnie mierzona po zerwaniu próbki, aby określić długość końcową w momencie zerwania L_u . Dystans L_0 jest



Rysunek 2.3: Kształt próbek o przekroju okrągłym: (a) przed testem; (b) po teście. [50]

najczęściej wielokrotnością 5 mm. Wydłużenie procentowe próbki A oblicza się z wzoru (2.1), a wynik bezwzględnej różnicy długości określa się z dokładnością do 0.25 mm [50].

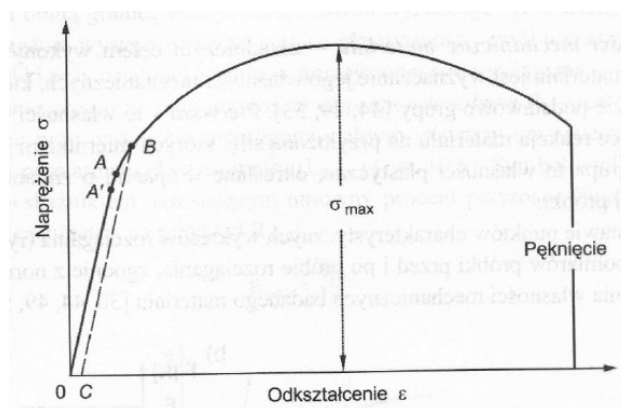
$$A = \frac{L_u - L_0}{L_0} \times 100 \quad (2.1)$$

Gdzie:

- d_0 - średnica początkowa próbki;
- L_c - długość obszaru o stałej średnicy ;
- L_0 - długość początkowa w obszarze pomiarowym;
- L_t - całkowita długość próbki testowej;
- L_u - długość końcowa próbki po teście;
- S_0 - przekrój początkowy próbki;
- S_u - przekrój końcowy próbki po teście.

Próbka jednoosiowego rozciągania, jak sama nazwa wskazuje, polega na rozciąganiu próbki w jednej osi oraz stałym zapisie osiowej siły rozciągającej w funkcji drogi trawersy maszyny wytrzymałościowej oraz w czasie. Wydłużenie próbki obliczone po teście może być zastosowane, wraz z zarejestrowaną siłą, do wykreślenia inżynierskiej krzywej rozciągania (Rysunek 2.4)

Na krzywej zobrazowana jest zależność odkształcenia ε od naprężenia σ , gdzie naprężenie zostało określone poprzez podzielenie wartości siły F w każdym momencie testu przez początkowy przekrój poprzeczny próbki S_0 2.2.



Rysunek 2.4: Inżynierska krzywa rozciągania w układzie naprężenie - odkształcenie.

$$\sigma = \frac{F}{S_0} \quad (2.2)$$

Kluczowym aspektem oznaczonym na krzywej (Rysunek 2.4) jest tzw. granica proporcjonalności, stanowiąca szczególne wyzwanie przy porównaniu wyników SPT i UTT. Wyznaczone tak naprężenie na wykresie zobrazowane jest jako granica prostoliniowego przebiegu (A') tj. proporcjonalnej zależności naprężenia od odkształcenia. Samo nachylenie krzywej w tym zakresie zależne jest od wartości modułu Young'a. Przebieg ten kończy się granicą plastyczności, która określana jest jako naprężenie, przy którym powstają nieodwracalne odkształcenia (Wzór 2.3). Najczęściej występującym typem granicy plastyczności w materiałach jest umowna granica plastyczności, gdzie moment przejścia w stan nieodwracalnych odkształceń nie jest wyraźnie widoczny na krzywej, a wyznacza się go graficznie (Rysunek 2.5) dla wartości trwałego odkształcenia równego 0.2%.

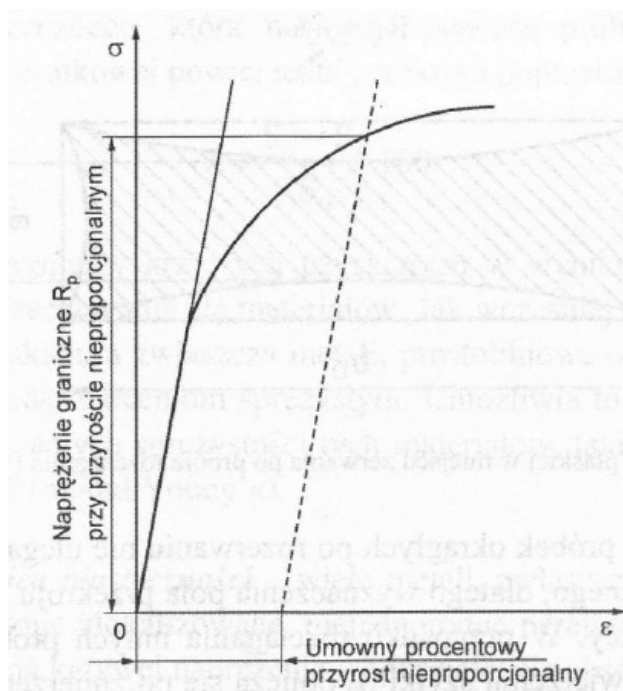
$$R_{p0.2} = \frac{F_{p0.2}}{S_0} \quad (2.3)$$

Dla zastosowań konstrukcyjnych ostatnim i najważniejszym parametrem wyznaczanym z pomocą próby jednoosiowego rozciągania jest wytrzymałość na rozciąganie, nazywana również granicą wytrzymałości R_m (Wzór 2.4). Wyznacza się ją, odczytując najwyższą wartość siły rozciągającej w trakcie testu, występującą tuż przed rozpoczęciem szybkowania, tj. momentem, zanim materiał utraci stateczność.

$$R_m = \frac{F_m}{S_0} \quad (2.4)$$

2.2.2 Badanie udarności

Metale mogą ulegać dwóm różnym rodzajom pęknięcia: kruchym lub plastycznym. Rozprzestrzenianie się uszkodzeń następuje szybciej w materiałach kruchych ze względu na zdolność materiałów plastycznych do ulegania odkształceniom plastycznym. W ten sposób materiały plastyczne są w stanie wytrzymać większe naprężenia, ze względu na ich zdolność do pochłaniania większej ilości energii przed zniszczeniem, niż materiały kruche. Odkształcenie plastyczne powoduje, że naprężenie pęknięcia wzrasta z powodu pracy plastycznej wymaganej do rozszerzenia pęknięcia, dodając do pracy wymaganej do utworzenia pęknięcia



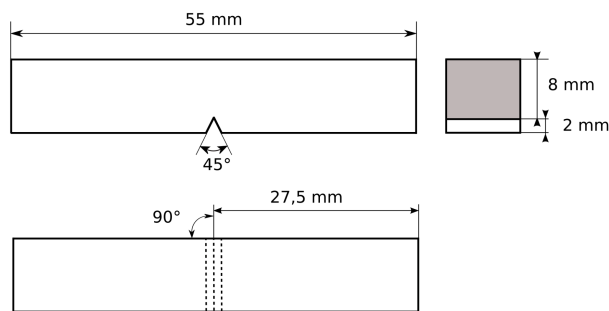
Rysunek 2.5: Zakres plastyczny na inżynierskim wykresie naprężenie-odkształcenie

- pracy, która odpowiada wzrostowi energii powierzchniowej wynikającej z utworzenia dodatkowej powierzchni pęknięcia. Odkształcenie plastyczne w metalach plastycznych jest ważne, ponieważ może być oznaką potencjalnego uszkodzenia metalu. Jednak punkt, w którym materiał wykazuje zachowanie plastyczne w porównaniu z zachowaniem kruchym, zależy nie tylko od samego materiału, ale także od temperatury, w której do materiału przykładane jest obciążenie. Temperatura, w której materiał zmienia się z kruchego na plastyczny lub odwrotnie, ma kluczowe znaczenie dla projektowania metalowych produktów przenoszących obciążenia dynamiczne. Minimalna temperatura, w której metal zmienia się z zachowania kruchego na plastyczne lub z zachowania plastycznego na kruche, znana jest jako temperatura przejścia plastyczno-kruchego (z ang. *Ductile-Brittle Transition Temperature - DBTT*). Poniżej DBTT materiał nie będzie w stanie odkształcać się plastycznie, a szybkość propagacji pęknięć gwałtownie wzrośnie, prowadząc do szybkiego uszkodzenia materiału. Ponadto DBTT jest ważna, ponieważ gdy materiał schładza się poniżej DBTT, ma znacznie większą tendencję do pęknięcia pod wpływem uderzenia, zamiast zginania lub odkształcania plastycznego (kruchosc w niskiej temperaturze). Zatem DBTT wskazuje temperaturę, w której zdolność materiału do odkształcania plastycznego zmniejsza się wraz ze spadkiem temperatury, a tym samym szybkość propagacji pęknięć dramatycznie wzrasta [141, 32]. Do określania udatności materiału wykorzystuje się w przemyśle dynamiczne testy łamania próbek, aby zarejestrować ich energię pęknięcia. Do ustandaryzowanej grupy testów należy próba Charpy'ego. W czasie testu zaabsorbowana energia podczas uderzenia jest miarą udatności materiału. Aby wyznaczać udatność, korzysta się z wzoru (2.5).

$$KC = \frac{K}{S} \left[\frac{J}{\text{mm}^2} \right] \quad (2.5)$$

Gdzie:

- KC - udatność;



Rysunek 2.6: Standardowe wymiary próbki do testu udarności [32].

- K - praca uderzenia [J];
- S - pole powierzchni przekroju próbki w miejscu karbu [mm^2].

Próbki wykorzystywane do testu są ustandaryzowane, co pozwala lepiej porównywać wyniki pomiędzy różnymi laboratoriami, ponieważ miara jest często komparatywna, a do określenia zgodności z wymaganiami, np. projektowymi, potrzebna jest wartość referencyjna. Zgodnie ze standardem [32] próbki mają wymiary 10x10x55 mm z głębokością wcięcia 2 mm (Rysunek 2.6).

Przebieg testu polega na łamaniu próbki, naciętej w taki sposób, aby powstał karb, pojedynczym uderzeniem wahadła. Próbką jest zamocowana w dwóch narzędziach, które wspierają ją podczas testu po drugiej stronie od miejsca uderzenia wahadłem. Test wykonywany jest przy różnych temperaturach, aby określić udarność próbki w zależności od wymaganego zastosowania dla danego materiału [32].

2.2.3 Statyczna próba pełzania

Aby określić wpływ stałych sił występujących w konstrukcjach, takich jak mosty czy zbiorniki ciśnieniowe, przeprowadza się badania pełzania materiału. Badanie, choć zunifikowane z założenia, jest ściśle zależne od założeń konstrukcyjnych oraz zastosowania danego materiału. Jego wytrzymałość na pełzanie determinuje się poprzez przyjęty cykl życia konstrukcji oraz obciążenia przyjęte na wstępnych etapach projektu. Próbki materiału poddawane są testom pełzania odpowiadającym rzeczywistym warunkom, jak i przyspieszonym testom pełzania oraz zadawaniu wzmacnianych obciążeń mechanicznych i termicznych. W obu przypadkach test może trwać setki, a nawet tysiące godzin. Próba jednoosiowego pełzania [19] wykonywana w warunkach statycznych polega na podgrzaniu próbki do określonej w zakresie testu temperatury oraz na zadawaniu odkształceń w próbce poprzez aplikowanie stałej siły rozciągającej lub napięcia rozciągającego wzdłuż osi wzdłużnej próbki przez określony czas, aż do uzyskania:

- określonego w ramach testu wydłużenia końcowego (nieprzerwany test);
- wartości określonego wydłużenia w każdym z interwałów testu (test przerywany);
- czasu pęknięcia przy pełzaniu (test przerywany i nieprzerwany).

Test wykonuje się na maszynie wytrzymałościowej, gdzie zadanie polega na obciążeniu próbki wzdłuż jej osi, minimalizując niepożądane momenty gięcia i skręcania poprzez poprawne i kontrolowane zamoco-

wanie próbki w polu testowym. Siła obciążenia próbki nie powinna być gwałtowna, aby ograniczyć dynamiczny aspekt testu, a maszyna testowa powinna być odizolowana od wpływu zewnętrznych wibracji.

2.3 Nieniszczące metody diagnostyki stanu materiału

SPT z zasady stanowi most łączący metody nieniszczące oraz konwencjonalne metody niszczące do badania stanu materiału. Na polu metod nieniszczących prym wiodą metody, których wyniki przedstawiają pośrednie informacje na temat parametrów określających stan materiału, znanych z metod konwencjonalnych. Już na wstępie stawia to metodę SPT w pozycji metody atrakcyjnej, uzupełniającej lukę widoczną na rynku komercyjnych badań konstrukcji będących w stałym użyciu.

2.3.1 Magnetic Barkhausen Noise (MBN)

Metoda analizy szumów Berkhausena (z ang. *Magnetic Barkhausen Noise (MBN)*) jest metodą nieniszczącą, wykorzystująca zjawisko szumów indukowanych w ferromagnetycznych materiałach w wyniku oddziaływania pola magnetycznego. Metodę opracował w roku 1919 Heinrich Berkhausen, jednak swoje przemysłowe zastosowanie znalazła dopiero we wczesnych latach 80 XX wieku [135, 59]. Metoda ta zapewnia informacje o zachowaniu materiału na jego powierzchni oraz do 1 mm tuż pod nią, jednak ma szerokie spektrum częstotliwości, aż do ponad 2 MHz dla większości materiałów ferromagnetycznych. Estymowana głębokość penetracji sygnału może być wyznaczona ze wzoru 2.6 [135]:

$$\delta = \frac{1}{\sqrt{\pi\mu\sigma f}} \quad (2.6)$$

gdzie δ to głębokość penetracji w [mm], μ przedstawia przenikalność magnetyczną, σ to przewodność elektryczna, a f to częstotliwość zmiennego pola magnetycznego.

Typowy układ pomiarowy zawiera:

- Rdzeń magnetyzujący w postaci cewki lub jarzma wytwarza zmienne pole magnetyczne
- Cewkę detekcyjną rejestrującą szumy Berkhausena
- Moduł analizy do obliczania parametrów takich jak:
 - Pierwiastek średniokwadratowy z wartości szumów
 - Liczbę impulsów
 - Energię sygnału
 - Pozycję maksimum szumu względem cyklu histerezy

Metoda znajduje ograniczenia już w wyborze materiału do zastosowania, ponieważ bez właściwości magnetycznych materiałów nie może być badany metodą MBN. Do najczęściej stosowanych materiałów należą stale, z wyłączeniem stali niemagnetycznych, takich jak austenityczne stale nierdzewne, żeliwa oraz niektóre stopy niklu i kobaltu. MBN jest również wrażliwa na zmiany mikrostruktury przy powierzchni, gdzie możliwe jest przeprowadzenie testu. Duża gęstość dyslokacji generuje rozproszenie sygnałów, ponieważ wraz z

jej wzrostem rośnie liczba domen magnetycznych. Znaczący wpływ mają również naprężenia szczątkowe występujące w materiałach odkształcanych, nienormalizowanych, które akumulują się przy powierzchni. Naprężenia rozciągające zwiększają wartości odczytywane w badaniu MBN, a ściskające zmniejszają dlatego istotnym aspektem jest dobór obróbki cieplnej [59].

Metoda znajduje zastosowanie przemysłowe przy badaniu konstrukcji będących w użytku ze względu na swój nieniszczący charakter i sprawdza się doskonale w przypadku analizy uszkodzeń, takich jak te powstałe przy szlifowaniu oraz obróbce powierzchniowej rurociągów i turbin. Pozwala ona na:

- Mapowanie naprężeń szczątkowych występujących przy powierzchni konstrukcji ze względu na ich czułość.
- Ocenę uszkodzeń zmęzeniowych
- Pomiar głębokości utwardzenia.

W uogólnieniu zapewnia ona kompleksową analizę powierzchni komponentów stalowych pod kątem jakości cieplno-mechanicznej. Nie zapewnia jednak informacji na temat głównych własności materiału, dlatego może być dodatkowo uzupełniana przez metodę SPT w zastosowaniach przemysłowych.

2.3.2 Badanie mikrotwardości

Metody z grupy badań mikrotwardości stoją w kontrze do metod klasycznego badania twardości, które z zasady są metodami niszczącymi ze względu na rodzaj wykorzystywanych próbek. Badanie mikrotwardości służy do określenia twardości i odporności na deformację tam, gdzie niemożliwe jest wykorzystanie metod klasycznych. Niezależnie jednak od skali badania i wielkości próbki, musi ona zostać pobrana z materiału bazowego, z konstrukcji pracującej lub wyłączonej z użytku. Rozróżnia się dwa warianty badań: metoda Vickersa oraz metoda Knoop [49].

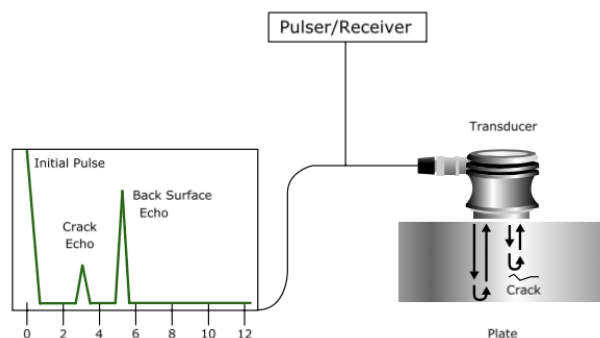
Metoda Vickersa (HV) wykorzystuje próbnik zakończony piramidalnym stożkiem o kącie rozwarcia 136 stopni, który, przy niewielkiej sile nacisku do 10 N, wgniata się w badaną powierzchnię przez 10 sekund. Po zakończeniu testu mierzy się przekątne wgniecenia. Metoda ta nie jest jednak aplikowana dla powłok o grubości poniżej 60 mikronów ze względu na możliwe różnice we właściwościach warstw przypowierzchniowych. Twardość wyznaczana jest ze wzoru 2.7 [52]:

$$HV = \frac{1.8544 \cdot F}{d^2} \quad (2.7)$$

gdzie: HV to twardość w skali Vickersa podawana w jednostkach $[kgf/mm^2]$, F to siła obciążenia w $[kgf]$, a d to średnia długość przekątnych odcisku w $[mm]$.

Metoda Knoop (HK) zasadniczo jest podobna do metody Vickersa, jednak wykorzystuje inny kształt stempla wgniatającego próbki. Jego kształt jest wydłużony w przekroju i pozostawia wgniecenie w kształcie diamentu lub, w kontekście figur płaskich, rombu. W metodzie Knoop do oceny rozmiaru wgniecenia mierzy się jedynie długość najdłuższej przekątnej. Wykorzystywana jest z obciążeniami mniejszymi niż 10 N, najczęściej przy wielokrotnych, sąsiadujących pomiarach oraz przy krawędziach próbek. Twardość wyznacza się ze wzoru 2.8 [51]:

$$HK = \frac{14.229 \cdot F}{L^2} \quad (2.8)$$



Rysunek 2.7: Uproszczony schemat wykorzystania UT do wykrywania pęknięcia w materiale [97]

gdzie: HK - twardość w skali Knoop, F - siła obciążenia w $[gf]$, a L - długość najdłuższej przekątnej odcisku w $[mm]$.

Metody te znajdują zastosowanie w badaniach:

- Powłok materiałów nawęglanych i azotowanych.
- Materiałów twardych i ceramicznych
- Spoin i stref wpływu ciepła (HAZ).

Wymogiem ograniczającym szersze zastosowanie badań mikrotwardości jest konieczność przygotowania powierzchni do stanu niemal lustrzanego, ponieważ ze względu na skalę badania wpływ niedoskonałości powierzchni może silnie zaburzać wyniki otrzymywane oraz interpretacje stanu materiału. Badania nie mogą być również wykonywane in-situ ze względu na czułość aparatury pomiarowej, dlatego klasyfikuje się je jako badania laboratoryjne. Występujące tu podobieństwo do metody SPT należy scharakteryzować szerzej, ponieważ obie metodologie wykorzystują wgniatanie próbek stemplem, jednak w przypadku badania SPT dokonuje się pomiaru przemieszczenia, a nie rozmiaru wgniecenia pozostałego w badanej próbce. Co więcej, metoda SPT w zakresie analizy pojedynczej próbki niszczy ją w zamian za uzyskanie wartości granicy wytrzymałości. Dla badań twardości wartości parametrów własności materiałów otrzymywane są jedynie z przybliżenia i kart korelacji stosowanych w przemyśle i nie dają one bezpośredniego wyniku, jak ma to miejsce w przypadku SPT. Obie metodologie badań mogą się wzajemnie uzupełniać i pozwalać na szerszy wgląd w stan materiału, potencjalnie wykorzystując tę samą, niewielką próbkę pobraną z materiału bazowego.

2.3.3 Metody ultradźwiękowe

Metody ultradźwiękowe UT (z ang. *Ultrasonic Testing*) używają energii ultradźwiękowej o wysokiej częstotliwości do wykrywania wad wewnętrznych materiałów, badania wymiarów grubości próbek oraz charakteryzacji materiału pod kątem jego zużycia. Obrazowe przedstawienie zastosowania metody widoczne jest na rysunku 2.7, gdzie w schemacie przyrządów badawczych występują nadajnik-odbiornik, przetwornik oraz rejestrator z wyświetlaczem. Badanie opiera się na generowaniu i detekcji fal mechanicznych, które przenikają przez materiał i oddziałują z jego strukturą wewnętrzną.

Fale mogą ulegać odbiciu, załamaniu, rozproszeniu i tłumieniu, a rejestracja ich zachowania pozwala na wszechstronne rozpoznawanie niejednorodności występujących w strukturze badanego materiału, takich jak nieciągłości w postaci pęknięć i rozwarstwień, wtrąceń i porowatości materiału, lokalnych zmian grubości ścianek, a pośrednio także mikrostruktury poprzez detekcje zmiany predkości fal. W trakcie badania rejestrowane są takie parametry jak czas przelotu fali TOF (z ang. *Time of Flight*), amplituda echa mówiąca o wielkości defektu, wartości tłumienia fali wskazujące na mikrostrukturalną degradację oraz prędkość propagacji fali, która zależy głównie od struktury krystalicznej materiału [69]. W badaniach UT stosuje się różne typy fal ultradźwiękowych, takie jak:

- Fale podłużne są stosowane w materiałach litych, gdzie cząsteczki materiału drgają równolegle do kierunku propagacji.
- Fale poprzeczne są stosowane do wykrywania nieciągłości poprzecznych, gdzie cząsteczki drgają prostopadle do kierunku propagacji fal.
- Fale powierzchniowe wykrywające pęknięcia powierzchniowe.

Metody UT klasyfikowane są jako całkowicie nieniszczące, a jednocześnie charakteryzują się głęboką penetracją materiału, w przeciwieństwie do wcześniej opisanych metod. Są bardzo czułe na wewnętrzne defekty i nieciągłości, a dla dokładności pomiarów dają możliwości zastosowania zaawansowanych technik UT oraz różnych rodzajów fal. Dzięki tak wielu możliwościom, UT zajmuje główne miejsce w zestawieniu metod nieniszczących w badaniach stanu materiału [12]. Nie jest to jednak metoda w pełni kompleksowa, ponieważ nie dostarcza ona informacji o parametrach wytrzymałościowych materiału. Sama w sobie jest też metoda skomplikowaną, wymagającą dużego doświadczenia operatora, zwiększającą wpływ czynnika ludzkiego na dokładność wyników, co uwiadamia się w przypadku złożonych geometrii badanych obiektów. W porównaniu do metody SPT, może jednak być stosowana bezpośrednio na badanym obiekcie, a niekoniecznie w laboratorium.

2.3.4 Metoda prądów wirowych

Badanie materiału z wykorzystaniem prądów wirowych ECT (ang. *Eddy Current Testing*) opiera się na zjawisku indukcji elektromagnetycznej, gdzie w cewce nadawczej przepływ prądu zmiennego generuje zmienne pole magnetyczne, które po zbliżeniu do powierzchni przewodnika indukuje w nim prądy wirowe, tworzące w materiale wtórne pola magnetyczne. To z kolei wpływa na impedancję cewki, dzięki czemu możliwe jest wykrywanie w strukturze materiału zmian w postaci pęknięć, które powodują zmiany w rozkładzie prądów wirowych, a co za tym idzie, zmianę sygnału pomiarowego. Podstawowe przyrządy stosowane do badania ECT składają się ze źródła prądu zmiennego, cewki połączonej ze źródłem prądu oraz woltomierza do pomiaru zmian napięcia w cewce. Wraz ze stopniem zaawansowania metod, modyfikacje oraz dodatkowe przyrządy są stosowane w celu dokładniejszej detekcji i usprawnienia metody [96].

Efektywna głębokość penetracji prądów wirowych zależy od częstotliwości pola i właściwości badanego materiału. Opisywana jest wzorem 2.9 [16] wskazującym na odwrotnie proporcjonalną zależność pomiędzy

częstotliwością a głębokością detekcji wad:

$$\delta = \sqrt{\frac{2}{\mu\sigma\omega}} \quad (2.9)$$

Gdzie: δ - głębokość penetracji [m], ω - częstotliwość kołowa [Hz], μ - przenikalność magnetyczna [H/m], a σ - przewodność elektryczna [S/m]. Metoda ECT najczęściej znajduje zastosowanie w wykrywaniu pęknięć powierzchniowych oraz w pomiarze warstw przewodzących, takich jak farby czy warstwy tlenków na powierzchni materiału. Jest ona niezwykle czuła na zmiany powierzchniowe i jest relatywnie prosta w użyciu, co przyczynia się do możliwości automatyzacji, np. na liniach produkcyjnych. Ograniczona jest jednak do materiałów przewodzących, a ze względu na punktowy kontakt badań, niewłaściwa orientacja przyrządu badawczego na powierzchni materiału może przyczynić się do niewykrycia wady. Dlatego znaczącym czynnikiem w dokładności badania jest doświadczenie operatora i właściwa kalibracja urządzenia z wykorzystaniem wzorców. W zestawieniu z metodą SPT, podobnie jak opisywane wcześniej rodzaje badań, nie pozwala na bezpośrednie wyciągnięcie z materiału jego parametrów wytrzymałościowych. ponownie jednak powyższy opis metody wskazuje na możliwość komplementarnego zastosowania metody SPT w połączeniu ze wszystkimi opisanymi do tej pory metodami, zwiększając wartość dostarczaną do zlecającego przez ośrodek badawczy prowadzący analizy.

2.4 Small Punch Test (SPT)

Prace nad korelacją wyników SPT i klasycznych badań jednoosiowego rozciągania z wykorzystaniem metod elementów skończonych (MES) prowadzone są od roku 2004. Przeprowadzono testy na miniaturowych próbkach ze stali austenitycznej AISI 316L [23, 70, 75] oraz martenzytycznej hartowanej 7-9Cr F82H [23]. Wykorzystano i zwalidowano tam modele opracowane specjalnie dla stali 316L. Badania te wykazały, że należy brać pod uwagę w obliczeniach i pomiarach stopień ugięcia maszyny pod wpływem całkowitego nacisku prasy. Dopiero w roku 2021 opublikowano badania traktujące o metodyce pomiaru, a następnie redukcji wpływu stopnia ugięcia maszyn testowych na wyniki testu [122]. Aby poprawnie wyznaczyć i zamodelować krzywe obciążenie-przemieszczenie dla SPT, należy uwzględnić również wpływ tarcia, który staje się bardziej widoczny w przypadku ugięcia przekraczającego 0,5 mm. Jego wpływ objawia się w badaniach jako zwiększoną siłą w odniesieniu do analiz nieuwzględniających tarcia. Ogólny przegląd powstawania krzywych SPT pozwolił na wyznaczenie etapów odkształcenia na wykresach. Na tym etapie jednak opierano się na wstępnych wersjach relacji wyników z testów SPT z metodami klasycznymi. Uzyskane wyniki pozwoliły na stwierdzenie braku bezpośredniej zależności granicy wytrzymałości od maksymalnego obciążenia zarejestrowanego w próbie, ze względu na różnice między stanem odkształcenia a obciążeniem przy maksymalnym obciążeniu, a także podczas szyjkowania w standardowej próbie. Do dalszych badań nad oceną korelacji wyników badania SPT i UTT wykorzystano stale 20MnMoNi55, CrMoV oraz SS304LN [71] oraz HR3C [76] Wyniki zestawiono z analizami elementów skończonych. Stwierdzono, że najlepsza zależność pomiędzy tymi dwoma metodami zachodzi, kiedy brany jest pod uwagę obszar szyjkowania, gdzie w analizie MES zakres przy przemieszczeniu 0.48 mm był najbliższym odpowiednikiem w próbie SPT. Podobnie, w kolejnym kroku [56] analizie numerycznej poddano stale niskostopowe SA508, Gr3,

TP316L, stale austenityczne oraz odlewane austenityczne stale stopowe CF8M. Dla poszczególnych materiałów otrzymano różne współczynniki α służące do konwersji wyników SPT na wartość DBTT uzyskaną w metodzie Charpy'ego. Analizą numeryczną, jak i klasyczną, poddaje się również próbki nacięte [87, 5], których zastosowanie opisane jest w aktualnej europejskiej normie EN 10371:2021. Ma na celu określenie parametrów pękania materiału oraz ustabilizowanie metody SPT jako alternatywy dla metod, w których nie jest możliwe wykonanie odpowiednio dużej naciętej próbki.

Jednym z kluczowych problemów związanych z badaniami SPT i im podobnymi jest rozmiar próbki do badań. Jest on znacznie mniejszy niż w przypadku standardowych badań materiałowych, co może powodować znaczne niepewności wyników. Zakładając jednorodność próbek, które mogą być wycięte z użytkowanych komponentów lub nowych stopów, celem jest pomiar własności na niewielkiej objętości materiału, tak aby otrzymane wyniki mogły być uzgodnione dla całej objętości. Możliwy wpływ niewielkiego rozmiaru próbki to [48]:

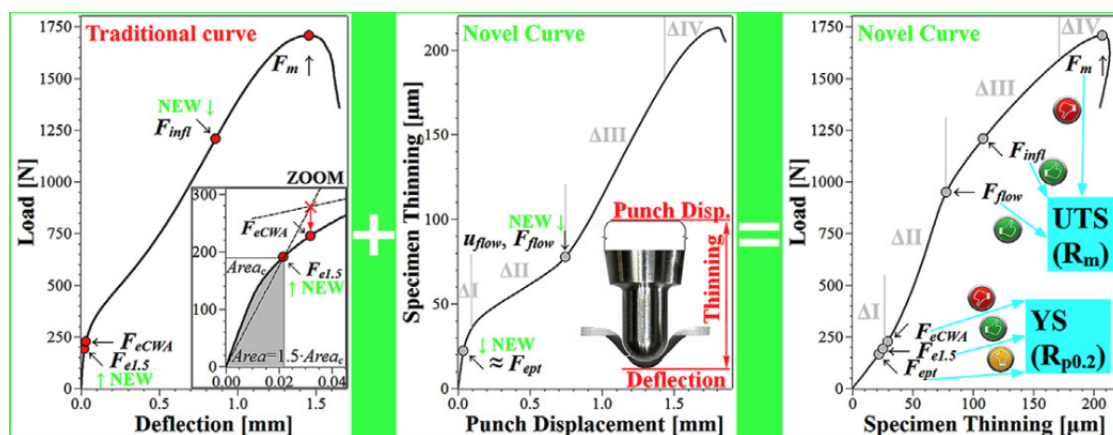
- Próbka zawiera niewystarczającą liczbę ziaren, które pozwoliłyby na odzwierciedlenie otrzymanych wyników w skali makro.
- Na mechanizm deformacji bezpośredni wpływ mają rozmiary ziaren oraz podobne regiony, takie jak granice ziaren martenzytu, kolonie bainitu oraz ich rozmiar.
- Modele deformacji i pęknięć są zależne od ilości małych defektów w materiale oraz ich wzrostu, jak ma to miejsce w przypadku statystycznego modelu Weibulla.
- Wpływ geometrii próbki, gdy zredukowana grubość prowadzi do odkształceń działających wzdłuż materiału, ogranicza płaski stan odkształcenia.
- Dodatkowe ograniczenia mogą występować przez niehomogeniczność, szczególnie przy regularnej formie, gdzie dane regiony mają ulepszone lub pogorszone własności wytrzymałościowe lub pełzania. Przykładem mogą być sekcje spawów, gdzie sprawy są umieszczone w strefie równoległej do długości próbki.

W pracy [48] przedstawiono wyniki badań, w których podjęto próbę numerycznego wyznaczenia i zdefiniowania reprezentatywnej objętości materiału (RVE, z ang. *Representative Volume Element*). Jest to najmniejsza objętość w modelu, w której obliczenia danego modułu są na tyle dokładne, że pozwalają na przedstawienie makroskopowej homogeniczności własności z wykorzystaniem indywidualnych, heterogenicznych, ale powtarzalnych elementów. Wynikiem badań był stosunek $RVE \geq$ do 20 rozmiarów ziaren kryształu w materiale, a wynik ten potwierdzono dla ponad 250 różnych materiałów. Aby wyeliminować obawy o niepewność otrzymanych wyników, należy zapewnić co najmniej 6-8 ziaren w przekroju poprzecznym próbki lub łącznie 200-500 cech charakterystycznych, takich jak ziarna i granice, w całkowitej objętości materiału w badanej próbce.

SPT jest stale rozwijającą się metodą, dlatego wykorzystano model uszkodzenia materiału GTN (Gurson-Tvergaard-Needleman) stworzony do opisu plastycznego pęknięcia materiału, przedstawiony w pracy [81]. Wsparcie opierało się na symulacjach MES, a wyniki uzupełniano przez wykorzystanie sieci neuronowych. Badanie SPT stwarza jednak dalsze problemy ze względu na niewielki rozmiar próbki oraz

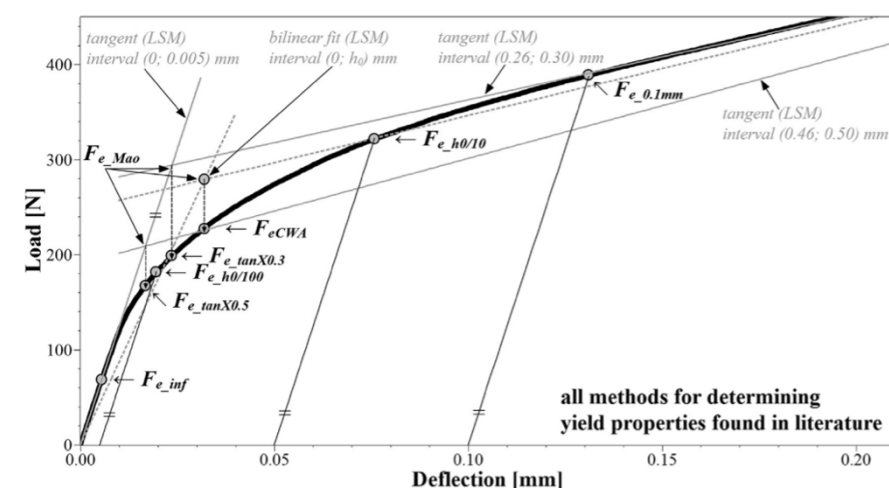
skomplikowany stan naprężenia. Badania nad stałą P91 [25] pozwoliły stwierdzić, że 65% energii plastyczności materiału jest tracone przed osiągnięciem maksymalnego obciążenia próbki. Z obserwacji mikrostruktury wynika, że znaczące pocienianie i szyjkowanie zachodzą dopiero po przekroczeniu maksymalnego obciążenia. Kolejne lata przynosiły kolejne zagadnienia potrzebne do rozwiązania przed komercyjnym zastosowaniem metody SPT, takie jak efekt Bauschingera. Badanie z pracy [22] wykazało, że jego obecność w próbce, tj. występowanie odkształceń plastycznych zaindukowanych naprężeniem plastycznym o znaku przeciwnym do tych indukowanych już w samej próbce SPT, uniemożliwia wiarygodne zastosowanie otrzymanych w teście wyników do obliczenia granicy plastyczności danego materiału.

W pracy [55] podjęto próbę obrania alternatywnej niż dotychczas ścieżki analizy wyników otrzymywanych w próbce SPT. Metodologia ta znacząco poprawiła wyniki badań. Do tej pory analizowano krzywe SPT jako obciążenie-przemieszczenie stempla lub obciążenie-ugięcie próbki. Nowatorska metoda polegała na połączeniu obu tych wykresów i przeprowadzeniu analizy obciążenia względem pocienienia próbki, uzyskanego poprzez odjęcie wartości przemieszczenia stempla od wartości ugięcia próbki (Rysunek 2.8).



Rysunek 2.8: Graficzne przedstawienie metodologii analizy krzywej obciążenie-pocienienie próbki. [55]

W eksperymencie zastosowano precyzyjne urządzenia pomiarowe z jednoczesnym pomiarem ugięcia oraz przemieszczenia, co pozwoliło na stały monitoring pocienienia próbki. Relatywnie duża objętość materiału znajdująca się bezpośrednio przed stemplem podlega uplastycznieniu już na samym początku próby, a przejście ze stanu sprężystego do plastycznego odbywa się w sposób ciągły. Całkowite uplastycznienie tej porcji materiału następuje znacznie wcześniej, niż zwykle stwierdza się to przy całej objętości. Na tej podstawie przedstawiono nowe kryterium wyznaczania granicy plastyczności materiału w badaniu SPT. Do tej pory stosowano siedem różnych podejść do określania wartości obciążenia odpowiadającej momentowi przejścia całej objętości badanego materiału w stan plastyczny, czyli momentu, w którym w całej jego objętości następują nieodwracalne odkształcenia (Rysunek 2.9). Co więcej, według tych badań, wszystkie charakterystyki materiałowe, normalnie otrzymywane w próbce jednoosiowego rozciągania, powinny być wyznaczone z krzywej jeszcze przed osiągnięciem maksymalnego obciążenia, gdyż wówczas materiał już całkowicie traci swoją stabilność. Zaproponowano zatem nowe podejście do analizy, jednak nie zostało ono przyjęte do obowiązującej normy. Należy zatem oczekiwać dalszego rozwoju tej metodologii dla szerszego spektrum materiałów.



Rysunek 2.9: Istniejące metody wyznaczania przejścia materiału w stan plastyczny w badaniu SPT. [55]

Odmienne niż wyżej wspomniane podejście zastosowano do syntezy obecnie stosowanych metodologii wyznaczania charakterystycznych własności mechanicznych różnych materiałów [8]. W tym przypadku zweryfikowano niepewność stosowanych podejść i określono rozrzut pomiędzy wynikami otrzymywanymi w badaniu SPT i klasyczną próbą jednoosiowego rozciągania. Analizie poddano stal T91 w dwóch wariantach: narażoną na napromieniowanie neutronami oraz kruchość, oraz w stanie pierwotnym. W eksperymencie pięć laboratoriów analizowało stal nienapromieniowaną, gdzie wygenerowano osiem grup danych dla dwóch podstawowych geometrii zestawów testowych. Relatywny rozrzut wyników dla energii całkowitej, obciążenia przy przejściu w stan plastyczny, obciążenia maksymalnego oraz przemieszczenia przy maksymalnym obciążeniu był analizowany dla badania w temperaturze pokojowej, jak i przy 300°C. Największy rozrzut zaobserwowano dla wartości siły w momencie przejścia materiału w stan plastyczny. Różnice wyników pomiędzy poszczególnymi laboratoriami były znacznie większe niż różnice w obrębie danego laboratorium, przy największych różnicach pojawiających się przy mniejszej geometrii próbki. Zależności określania własności mechanicznych były głównie zależne od danego urządzenia, na którym przeprowadzano próbę. W drugiej części badania, w laboratorium wiodącym, obserwowano próbki napromieniowane. W tym przypadku sam efekt napromieniowania próbek objawiał się na krzywych jako zwiększone obciążenie przy przejściu w stan plastyczny i zmniejszone maksymalne przemieszczenie bez znacznego wpływu na maksymalne obciążenie. Efekt ten był niezauważalny przy niższych temperaturach badania. Ogólnie przyznano dużą zgodność wartości temperatury przejścia w stan kruchy pomiędzy klasyczną próbą Charpy'ego a badaniem SPT w ujemnych temperaturach. Samo badanie SPT doskonale nadaje się do monitorowania zmian w wartościach DBTT przy materiałach napromieniowanych i kontroli pracujących komponentów pod tym względem. Efekt napromieniowania był także widoczny przy wartości umocnienia w porównaniu do wyników prób standardowych. Jednak wszelkie porównania pomiędzy wynikami w obrębie SPT powinny odbywać się na tym samym oprzyrządowaniu, gdyż wyniki zależą znacząco od użytego sprzętu [20].

Ogólna analiza zależności pomiędzy standardowymi testami, a metodami Small Punch przeprowadzona w pracy [21] jeszcze przed publikacją normy europejskiej. Podsumowano w niej dotychczasowe postępy w

systematyzowaniu metod badania właściwości materiałów. W ramach systematycznej analizy MES przygotowano 472 krzywe SPT, do których zastosowano następujące metody:

1. $Slope_{ini}$ - moduł Younga;
2. $Slope_{UL}$ - moduł Younga;
3. *Metoda Mao* – granica plastyczności;
4. *Metoda CEN's* – granica plastyczności;
5. *Metoda przesunięcia $t/10$* – granica plastyczności;
6. $Slope_{ini}$ – granica plastyczności;
7. *Metoda intersekcji* – granica wytrzymałości;
8. $Slope_{min}$ – granica wytrzymałości.

Wszystkie te metody oceny stosowane w SPT zależą od więcej niż jednego parametru materiałowego. Najlepsze rezultaty dla oceny granicy plastyczności otrzymano dla metody przesunięcia $t/10$, jednak ze względu na łatwość stosowania ulepszona metoda $Slope_{ini}$ jest praktyczniejsza. W przypadku granicy wytrzymałości metoda $Slope_{min}$ wykazuje mniejszy rozrzut wyników niż metoda intersekcji [21].

Wraz z badaniem metody z zastosowaniem eksperymentów fizycznych, rozwijał się także aspekt symulacyjny. Wielokrotnie wspomniane w niniejszej pracy symulacje z wykorzystaniem metod elementów skończonych są stale udoskonalane w kierunku jak najdokładniejszego odzwierciedlenia rzeczywistego przebiegu testów SPT. Praca [74] pokazuje dane własności mechanicznych w badaniu SPT odpowiadających tym otrzymywanym w próbie jednoosiowego rozciągania. Aby to osiągnąć, wykonano badania materiałowe klasyczne oraz SPT na 9 stopach metali. Z otrzymanych krzywych rozciągania wyznaczono parametry prawa Swifta (K, ϵ_0, n) będące stałymi indywidualnymi dla każdego materiału i użyto ich jako dane wsadowe do modeli elementów skończonych, aby wygenerować krzywe SPT dla badanych materiałów. Zależność ta została wygenerowana na podstawie 5 z 9 stopów stali będących grupą kalibracyjną. Krzywe rozciągania wygenerowano na podstawie krzywych SPT 4 z 9 stopów będących grupą testową poprzez metodę inwersyjną. Metoda ta polega na zbudowaniu bazy danych około 1000 hipotetycznych krzywych SPT wygenerowanych metodami elementów skończonych, używając krzywych naprężenie-odkształcenie reprezentujących pełne spektrum wartości parametrów (K, ϵ_0, n) . Krzywe z bazy danych, które najlepiej pasują do 4 materiałów z grupy testowej, są identyfikowane, a wartości (K, ϵ_0, n) są dopasowywane do generowania dokładniejszych wyników. Proces ten jest powtarzany aż do uzyskania zadowalających wyników. Po zakończeniu dopasowania, parametry (K, ϵ_0, n) używane są do wygenerowania krzywej rozciągania dla każdego z materiałów.

W przedstawionej historii badań i rozwoju metody SPT wyraźnie widoczna jest nisza pozostawiona przez badaczy pod kątem automatyzacji oceny stanu materiału i praktycznego podejścia do łączenia metody SPT z UTT. Metoda jest wciąż rozwijana, jednak kierunki przedstawiane w literaturze oraz na konferencjach

naukowych wskazują na próby alternatywnych podejść praktycznych oraz poszerzania dokładności symulacji, pozostawiając pole do badań nad przełożeniem badania na warunki praktyczne i weryfikację degradacji stanu materiału.

2.5 Small Punch Creep Test (SPCT)

Dobór stali do zastosowania w elektrowniach opiera się w dużej mierze o założenie, że dany materiał charakteryzować się będzie określoną odpornością na pełzanie. Niektóre testy mogą trwać nawet 30000 godzin, więc aby przyspieszyć wieloletnie analizy, coraz częściej stosuje się ekstrapolacje celem przybliżenia pożądaných wartości. W porównaniu do konwencjonalnych metod badania pełzania, SPCT stwarza znacznie więcej problemów i jest trudniejszy do wykorzystania w celu otrzymania własności pełzania danego materiału ze względu na złożone stany deformacji i rozkładu naprężeń w miniaturowej próbce materiału [15]. Mimo że pełzanie w próbce SPCT było analizowane na przestrzeni lat, większość badań skupiała się na geometrii próbki, a nie na pozostałych parametrach procesu. Po raz pierwszy w pracy [26] podjęto próbę skupienia się na innych parametrach poprzez rozwój nowej metody analizy z wykorzystaniem metody elementów skończonych. Z pomocą MES przemieszczenie i prędkość przemieszczenia zostały otrzymane dla różnych poziomów obciążenia. Związek pomiędzy naprężeniem i odkształceniem pełzania w próbce oraz zadaniem obciążeniem i wartością przemieszczenia został otrzymany z analizy wymiarowej i modelu rozciągania membrany. Po zakończeniu eksperymentów porównano wyniki otrzymane w standardowej próbce pełzania oraz w badaniu SPCT i zauważono dobrą zbieżność wartości w akceptowalnych granicach dokładności. To wskazuje na możliwość otrzymywania własności pełzania z wykorzystaniem badania SPCT zamiast badania klasycznego. Uzasadnieniem tej tezy może być fakt, że eksponenta pełzania może być otrzymana bezpośrednio z badania SPCT i jest ona zgodna z tą otrzymaną w standardowym badaniu.

Interpretacji wyników badania Small Puch Creep Test dokonano w pracy [47] skupiając się na weryfikacji i bezpośrednim zestawieniu wyników tej metody z klasycznym badaniem pełzania. Dokonano wtedy przedstawienia bezpośredniej zależności naprężenia w jednoosiowej próbce pełzania UCT oraz obciążenia zadawanego w badaniu SPCT. Poprzez wzór (2.10).

$$\sigma_{\text{ref}} = 0.512P \quad (2.10)$$

Gdzie: P [N] jest obciążeniem w badaniu SPCT, a σ_{ref} [MPa] naprężeniem w badaniu UCT. Ta zależność pozwala bezpośrednio interpretować wyniki w jednym badaniu i przeliczać je na ich odpowiedniki w drugim. Wymaga jednak korekcji i weryfikacji dla badania materiałów kruchych. Zależność ta była podstawą do znormalizowanego przelicznika stosowanego aktualnie w badaniach. W opisywanym eksperymencie kolejnymi wynikami analizy było stwierdzenie, że minimalna prędkość przemieszczenia w SPCT odnosi się do pewnej pozycji w trzeciorzędowym obszarze pełzania przy próbce UCT, jednak nie bezpośrednio do minimalnej prędkości pełzania w tejże próbce. Dalsze analizy potwierdziły tę zależność.

Zastosowanie badania SPCT ma miejsce w analizach materiałów podatnych na anizotropię własności materiału ze względu na metodę wytwarzania. Jedną z bardziej skomplikowanych metod wyciskania metali, zaliczaną przy tym do zaawansowanych metod obróbki plastycznej, jest wyciskanie przez kanał kontowy

ECAP (ang. Equal-channel angular pressing). Metoda ta stosowana jest głównie przy produkcji wyrobów ze stopów aluminium. Przeprowadzono badania materiałowe SPCT na wyrobach wyprodukowanych metodą ECAP, aby określić wpływ anizotropii na własności pełzania wyrobu gotowego [17]. Badanym materiałem był stop aluminium AL-Al4C3, a proces produkcji odbywał się w temperaturze pokojowej z wsadu wykonanego metodą metalurgii proszków. Wyciskanie materiału wykonano w dwóch przejściach schematem C tj. próbki były odwracane o 180° po pierwszym przejściu. Próbki do badań pobrano tak, aby kierunek siły i oś wyciskania zawierały między sobą kąty od 0 do 90° . W rezultacie uzyskane wyniki wskazywały jednoznacznie na różnice w wartościach dla próbek zależnie od pierwotnego kąta umiejscowienia próbki w wyciśniętym wyrobie. Pęknięcia miały międzykrystaliczny charakter, a luki po pęknięciach układały się współosiowo z kierunkiem wydłużenia. To badanie dowodzi użyteczności metody SPCT w komercyjnych zastosowaniach w przemyśle.

Kolejne próby badania zależności metody SPCT i standardowej próby jednoosiowego pełzania miały miejsce w pracy [95], gdzie badania prowadzono na stalach ferrytycznych o wysokiej zawartości chromu F82H MOD3. Skupiono się na wpływie atmosfery ochronnej na przebieg procesu i wyznaczenie parametru wiążącego próbę SPCT i próbę klasyczną. Jako atmosfery ochronnej używano argonu oraz próżni (poniżej 20^{-3} Pa). W badaniu SPCT, czas do pęknięcia, przebieg ugięcia próbki, minimalna prędkość ugięcia oraz ta w momencie pęknięcia próbki różniły się zależnie od użytej atmosfery ochronnej. Związek pomiędzy obciążeniem F [N] w SPCT i standardową próbą pełzania σ [MPa] przy takim samym czasie do pęknięcia również był różny w zależności od atmosfery ochronnej. Współczynnik α w równaniu 2.11 konwersji tych parametrów dla argonu wynosił 2.1, a dla próżni 2.3.

$$F = \alpha \sigma \quad (2.11)$$

Zasugerowano tu możliwy wpływ tarcia pomiędzy próbka i kulka przenoszącą obciążenie, powodujący różnice pomiędzy wynikami dla dwóch atmosfer ochronnych. Działanie tarcia potwierdzono w pracy [65] proponując alternatywną metodologię podejścia do konwersji wyników SPCT do UCT, która, jak pokazano w cytowanej pracy, wymaga dalszych badań.

W roku 2016 przedstawiono nowe podejście do określenia związku pomiędzy SPCT i UCT przez współczynnik k_{sp} [148]. Ponieważ stany naprężenia różnią się pomiędzy metodami, prowadzi to do różnic w rezultatach tych testów. Stan odkształcenia i naprężenia w badaniu SPCT należy przekonwertować na odpowiadający klasycznej próbie pełzania. Liczne badania przyczyniły się do ustanowienia tych związków, w tym najznajmniejsze, ujęte w normie, wykonane przez Chakrabarty'ego. Przedstawione w artykule podejście składa się z trzech kroków: wykonania obu badań na tym samym materiale, wyznaczenia odkształcenia i naprężenia z krzywej SPCT, wybrania modeli pełzania dla różnych wartości k_{sp} i porównania wyników z SPCT z tymi z klasycznej próby oraz wyznaczenia najlepszego dopasowania k_{sp} . Do badań wykorzystano stale P91, 1.25Cr0.5MoSi oraz 16Mo3. Analiza wyników wykazała podobieństwo krzywych ze SPCT oraz z próby jednoosiowego pełzania, gdzie dla SPCT wydzielić można trzy etapy pełzania. Również ta sama wartość stacjonarnego pełzania została zaobserwowana dla SPCT. Wykazano tę samą zależność minimalnej prędkości odkształcenia od naprężenia oraz czas do pęknięcia, co wskazuje na możliwość zamiennego stosowania obu metod badawczych. Nowe podejście, jakie zasugerowano, wykorzystuje wspomniany wcześniej

współczynnik k_{sp} , który jest parametrem materiałowym i może różnić się od wartości wstępnie przyjętych. Oznacza to, że aby zastosować to podejście dla nowego materiału, należy najpierw zweryfikować go eksperymentalnie.

Holmström, Dymacek i inni [43] podjęli próbę weryfikacji metodologii estymacji parametrów pełzania z wykorzystaniem SPCT i ich konwersji na dane otrzymywane w standardowym teście. Zauważyli oni, że modele zawierające ugięcie u i jego prędkość $\dot{u}$ w czasie próby jako czynniki zmienne dają znacznie dokładniejsze przełożenie niż klasyczne modele CSW ze stałym czynnikiem w konwersji obciążenia przy próbie SPCT do naprężenia w UCT. Główny rozrzut obliczonych wyników jest powodowany przez niepewności w określaniu wyżej wymienionych parametrów oraz przedwczesne pękanie. Testy prowadzono na stali P92, która wykazywała problemy z plastycznością i przedwczesne pękanie w temperaturach 600°C i 625°C w próbach SPCT. Autorzy uważają, że nowo zaproponowana metodologia zatwierdzona w normie europejskiej daje poprawne wyniki dla oceny parametrów pełzania dla materiałów plastycznych.

Duża czułość na zmiany charakterystyki materiałowej i mikrostrukturalnej badanych próbek potwierdzono w analizie [123] stali ASTM A387 5Cr-0.5Mo, na których przeprowadzono badanie SPCT w 550°C i 650°C. Zmiany mikrostrukturalne powiązano ze zmianami własności pełzania. Zaobserwowano, że starzenie stali przy 600°C przez 6000 h i 7000 h zainicjowało wytworzenie się węglików M₂₃C₆, które wpływały na krótszy czas do pęknięcia w stosunku do stali w stanie wyjściowym.

Mimo, iż badanie SPCT szeroko stosowane jest już od 2006 roku dzięki CoP (ang. European Code of Practice) to wciąż przed i po publikacji normy wyniki są porównywane i przeliczane na ich odpowiedniki w standardowej próbie UCT. Alternatywne podejście zaproponowano w pracy [149], mające polegać na ocenie własności pełzania badanych materiałów tylko na podstawie wyników SPCT. Badanie przeprowadzono dla stali P91 w temperaturze 600°C przy różnych obciążeniach. Rezultaty analiz potwierdziły tezę, że metoda pozwala na wyznaczenie pożądaných cech. Wyniki mieściły się w granicach błędu, a dodatkowo wzięto pod uwagę efekty takie jak odkształcenie plastyczne zachodzące już na początku obciążenia na powierzchni styku próbki ze stemplem. Wyniki weryfikowane były symulacjami komputerowymi, jednak zaznaczono, że problemem przy szerszym zastosowaniu i walidacji metody, czynnikiem ograniczającym może być koszt i czas wspomagających symulacji komputerowych.

2.6 Inne wybrane parametry materiałowe

Trwające badania ukierunkowane są na poszerzanie spektrum otrzymywanych z SPT parametrów materiałowych. Stopień zaawansowania i dokładności formuł stosowanych do ich wyznaczania wskazuje na potrzebę dalszego doskonalenia metod zalecanych przez obowiązujące standardy.

2.6.1 DBTT

Już w latach 80-tych zauważono, że wartości temperatury przejścia w stan kruchy otrzymywane przy badaniach SPT są znacznie niższe niż w standardowej próbie [94]. Wymagało to wypracowania jasnej zależności pozwalającej na przeliczenie wyników temperatury przejścia w stan kruchy danego materiału (DBTT) otrzymanych w badaniach SPT na uniwersalnie stosowane wartości. Przyjęto wtedy i opracowano model dla

stali, którego wynikiem była wartość współczynnika $\alpha = 0.35$ wyznaczona z wzoru (2.12), dalej potwierdzona w późniejszych latach [63, 41]. Współczynnik stosuje się do wyznaczania temperatur podanych w Kelvinach. Wartość ta jest zbliżona do obecnie unormowanej wartości 0.42 i 0.38 dla kolejno standardowej i miniaturowej próbki. Należy jednak brać pod uwagę adiabatyczny wzrost temperatury generowany przy dynamicznym odkształceniu. Zwiększa to temperaturę próbki, dlatego znacznie dokładniejsze wyniki otrzymać można przy statycznych próbach, gdzie prędkość posuwu stempla wynosi mniej niż 0.5 mm/s. Niemniej jednak badanie SPT jest mniej czułe na wpływ zmian temperatury niż standardowa próba Charpy'ego [88]. W kolejnych pracach [66] poprzez badania stali 14MoV63 pochodzących z rurociągów, gdzie materiał pracował przez 100000 h w temperaturze 520°C, potwierdzono zależność pomiędzy temperaturą nagłego spadku plastyczności a temperaturą przejścia w stan kruchy otrzymywanej w konwencjonalnym teście, co potwierdza zdadność metody SPT do użytku w celu określenia czułości na formowanie kruchych pęknięć w badanych materiałach dla danych temperatur eksploatacji.

$$\frac{T_{sp}}{FATT} = \alpha \quad (2.12)$$

Gdzie:

- T_{sp} - temperatura charakteryzująca punkt przejścia w stan kruchy w badaniu SPT [K];
- $FATT$ - temperatura przejścia w stan kruchy w badaniu materiału metodą Charpy'ego [K].

Gwarancja wysokiego bezpieczeństwa pracy w elektrowniach węglowych i atomowych wymaga spełnienia wysokich wymogów stawianych użytym materiałom konstrukcyjnym. Ocena zmian własności tych materiałów w trakcie ich eksploatacji jest do tego kluczowa. W przypadku elektrowni atomowych realnym ryzykiem jest kruchość stali spowodowana silnym napromieniowaniem. Aby nie wyłączać reaktorów z użytku, możliwe jest pobranie miniaturowej próbki do badań SPT. Dla tych konkretnych zastosowań SPT jest przydatną metodą, gdyż wykorzystuje dwuosiowy stan naprężenia, który także występuje na ściankach zbiorników stosowanych w elektrowniach.

W zakresie przejścia materiału w stan kruchy, stale ferrytyczne zawodzą ze względu na transkrystaliczne lub międzykrystaliczne pęknięcia powstające przy niedoskonałościach materiału zapoczątkowanych mikroskopijnymi odkształceniami plastycznymi. Aby dokładnie określić mechanikę pęknięcia, należy zidentyfikować parametry opisujące utwardzanie i mięknięcie materiału w postaci krzywych rzeczywistego naprężenia do rzeczywistego odkształcenia. W testach SPT krzywa naprężenie – ugięcie może być podzielona na charakterystyczne obszary reprezentujące zakres sprężysty, plastyczny, umocnienie i własności mechaniczne przy pęknięciu.

W badaniach materiałów za pomocą SPT stosuje się wyznaczanie DBTT z wykorzystaniem naciętych próbek. Głębokość nacięcia została przeanalizowana i w pracy [146] podjęto próbę jej optymalizacji pod względem powtarzalności, dokładności i miarodajności otrzymywanych wyników. Głębokość wcięcia ma znaczący wpływ na trójosiowość naprężenia. Zbyt płytkie lub zbyt głębokie nacięcie uniemożliwia określenie TSP w danym zakresie. W wyniku analiz w eksperymencie stosunek głębokości nacięcia do grubości próbki $a/t = 0.3$ został zarekomendowany jako referencyjna wartość dla dalszych analiz.

2.6.2 Wydłużenie

Do tej pory nie określono jednoznacznej zależności wyników otrzymywanych w metodach SPT z wartością wydłużenia otrzymywaną w standardowych próbach rozciągania. Jest to skomplikowana zależność ze względu na różnice w stanach naprężenia występujące w obu metodach. Niemniej jednak poszukiwania tej zależności trwają już od przeszło dekady. Pierwsze próby wykazania zależności wartości siły i przemieszczenia w badaniach SPT zaproponowano w pracy [119] oraz zweryfikowano w [35]. W zależności ukazanej w równaniu (2.13) zestawiono maksymalne przemieszczenie stempla $d_{\max}$, wstępną grubość próbki materiału t oraz stałą $\gamma = 7$ będącą charakterystycznym parametrem materiału, aby otrzymać odpowiednik wydłużenia e wyznaczany w próbie jednoosiowego rozciągania. Badania prowadzono w strefach wpływu ciepła HAZ (ang. Heat Affected Zone) w połączeniach spawanych. Uwzględnienie grubości próbki w równaniu było krokiem naprzód w stosunku do pierwotnego równania zawierającego jedynie połączenie z maksymalnym przemieszczeniem stempla.

$$e(\%) = \gamma \cdot \frac{d_{\max}}{t} \quad (2.13)$$

Gdzie: $d_{\max}$ - maksymalne przemieszczenie stempla [mm]; γ - stała charakterystyczna materiału; t - początkowa grubość próbki [mm].

Równanie to nie zostało jednak zmienione, a otrzymywane wyniki nie były doprecyzowane w zestawieniu z wartością wydłużenia w standardowym badaniu jednoosiowego rozciągania od czasu jego pierwszego przedstawienia, w pracy [34], gdzie dodano dodatkowy parametr. Odnosił się on jednak jedynie do badanych w eksperymencie materiałów, a wzmianki o wyznaczaniu parametru wydłużenia nie ma również w aktualnej europejskiej normie. Wskazuje to na brak silnej zależności tego parametru z danymi uzyskiwanymi w próbie SPT, która w tym przypadku daje przybliżone wyniki.

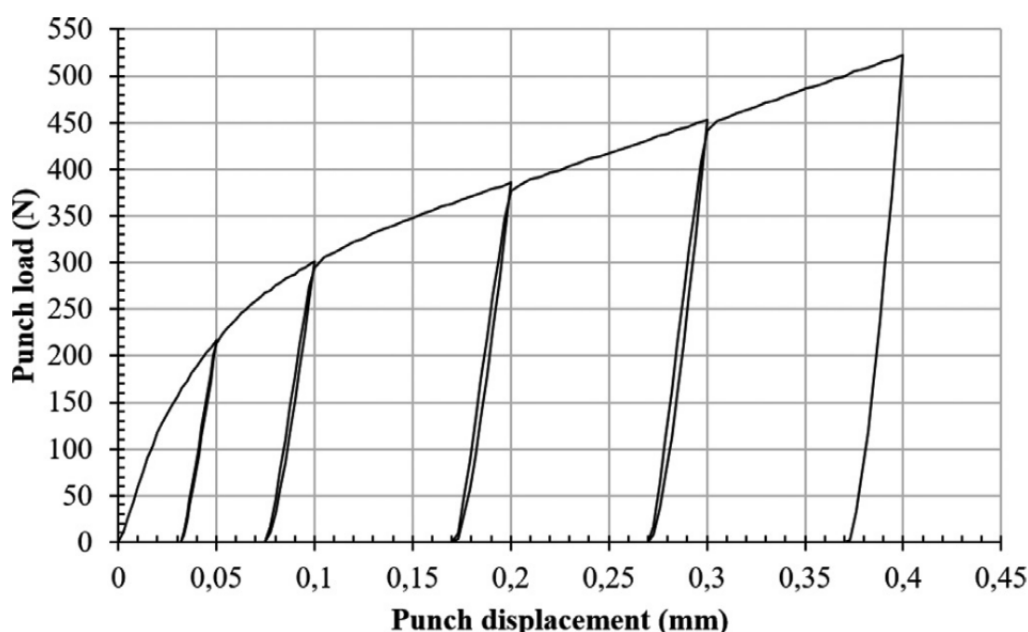
2.6.3 Moduł sprężystości

Wyznaczanie modułu sprężystości z wykorzystaniem badań SPT nie jest częstą procedurą i nie zostało również uregulowane. Niemniej jednak pewne zależności były poszukiwane już od powstania tej metody badawczej i z czasem udoskonalane. Tak też w pracy [27] zaproponowano nowe podejście do tego tematu. W eksperymencie użyto modelowania metodami elementów skończonych. Z wykorzystaniem obliczeń metodami elementów skończonych, analizy teoretycznej oraz eksperymentów. Korelacja ta cechuje się dużą dokładnością w określaniu wartości modułu sprężystości Younga E [GPa]. W bazowym założeniu do określenia wartości E wykorzystuje się nachylenie krzywej generowanej na początku obciążenia. Faktyczne zachowanie strefy sprężystej i standardowe nachylenie w tym obszarze może być reakcją sztywności badanej próbki i jej ugięcia oraz sztywności stempla wgniatającego próbkę. Ta kombinacja generuje mniejszą sztywność niż czyste sprężyste zginanie. Dlatego też dwa materiały o takim samym module sprężystości, a innej granicy plastyczności, wygenerują inne nachylenia krzywej w początkowym etapie obciążenia. Te obserwacje wskazują, że metodyka stosowana do tej pory zawiera błędy i pozostawia pole do udoskonalenia. Z tego względu podjęto próby analizy symulacyjnej i wprowadzono nowy sposób obciążania, aby wygenerować miarodajne wartości nachylenia. Proces obciążania próbki w próbie SPT podzielono na cykle

obciążania i odciążania (Rysunek 2.10). Tak uzyskane wartości nowego czynnika, jakim jest $Slope_{UL0.1}$ zastosowano w równaniu (2.14). Gdzie współczynnik korelacji nazwany λ' jest niezależny od plastycznych właściwości materiału i powiązany jest jedynie z geometrią próbki i zestawu testowego. Tak też, idąc w kierunku standaryzacji testu i jego unormowania, przyjąć można wartość $\lambda_{UL0.1}^{FEM} = 14.282^{-1}$ dla grubości próbki równej 0.5 mm, średnicy stempla 2.5 mm, średnicy wewnętrznej dolnej matrycy równej 4.0 mm i promieniu zaokrąglenia 0.5 mm.

$$E = \lambda_{UL0.1}^{FEM} \cdot Slope_{UL0.1} \quad (2.14)$$

Tak dobrane wartości zapewniły standardowe odchylenie wyników na poziomie 3.5% dla symulowanych materiałów oraz od 5 do 10% dla materiałów rzeczywistych.



Rysunek 2.10: Symulacje MES cykli obciążania i odciążania próbki. [27]

2.7 Zastosowanie metody SPT dla różnych materiałów

Większość badań metodyką SPT oraz SPCT odbywa się na materiałach metalicznych, a najczęściej na stalach. W literaturze znaleźć można wiele odniesień do badań właściwości mechanicznych stali w różnych stadiach wykorzystania przemysłowego, napromieniowania czy też takich z wyczerpaną żywotnością [94, 41, 121, 127]. Badania na stalach 15Ch2MFA, O8Ch18NT10, 10GN2MFA oraz A533JRQ w stanach przed i po obróbce termicznej i plastycznej w pracach [128, 129] weryfikowały dokładność metody SPT. Rezultaty pozwoliły zwiększyć przydatność tej metody w zastosowaniach komercyjnych i zaobserwować, jak czuła jest ta metoda na wszelkie zmiany w materiałach. Mając odpowiednio dużą bazę danych, można rozpoznawać procesy termoplastyczne w materiałach i ich wpływ na właściwości mechaniczne. W tym zakresie jednak wciąż wymagane są badania mające na celu zmniejszenie odchyłek, usystematyzowanie i ugruntowanie SPT jako standardowej metody badań materiałów. Dopiero z tego miejsca będzie możliwa ekspansja na

bardziej wymagające i zmienne obszary materiałów niemetalicznych. Niemniej jednak obiecujące postępy w tym kierunku zostały już poczynione. Przykłady tych działań znaleźć można w wielu pracach.

Podjęto próbę analizy wytrzymałości na pękanie czterech stopów aluminium z wykorzystaniem metody SPT [147]. Dane wartości odpowiadające rzeczywistym wynikom zestawiono z krzywymi otrzymanymi poprzez symulację badania SPT w oprogramowaniu ABAQUS. Otrzymane wartości krytycznej gęstości energii odkształcenia zestawiono z obciążeniami, kiedy obserwowano inicjację pęknięcia. Nie znaleziono jednak korelacji pomiędzy tymi dwoma parametrami. Niemniej jednak, wykorzystanie symulacji danego procesu na zasadzie metod elementów skończonych jest częstą praktyką, zwłaszcza na wczesnych etapach rozwoju danej metody badań. Właściwe odwzorowanie zachowania materiału w określonych warunkach zapewnić może wymierne wyniki odpowiadające fizycznym eksperymentom. W ten sposób, przy ograniczonych środkach i dostępie do szczególnych materiałów, możliwe stają się badania prowadzące do walidacji metody.

Kolejną z prób zastosowania metody SP do badań materiałów niebędących stopami stali było badanie SPCT w pracy [153] oraz starsze badanie SPT w pracy [85] gdzie analizie poddano jej wykorzystanie do określenia własności mechanicznych kompozytów w osnowie metalicznej LC4/SiCp. Do badania przyjęto stop bazowy LC4 oraz kompozyty o zawartości odpowiednio 7 i 14% SiCp. W badaniu zaobserwowano okrągłe pęknięcie, którego propagacja nastąpiła w bazowym stopie LC4. Zauważono również jednak wiele mniejszych pęknięć, rozwijających się w samym kompozycie w osnowie metalowej. Wnioskiem z badania było stwierdzenie, że wraz z umocnieniem od cząstek kompozyt wykazuje znacznie większą początkową sztywność, która ustępuje po powstaniu pierwszego pęknięcia. Podobnie w pracy [73] przeprowadzono analizę materiałową stopu tytanu Ti-6AL-4V odlewanego, a następnie przeciskanego izostatycznie, wykutego oraz wyprodukowanego metodą EBM (ang. Electron Beam Melting).

W literaturze można znaleźć wiele prób zastosowania metody SPT do analizy materiałów niemetalicznych. Przykładem takich działań może być analiza własności do deformacji tkanki kostnej w pracy [130], czy też badania cementu kostnego ze szkła akrylowego PMMA, od ang. poly(methyl methacrylate) przeprowadzone w pracy [38] oraz analiza cementu kostnego z materiału Palacos R® w pracy [28]. Metoda SPT okazała się być wysoce powtarzalna i efektywna do ewaluacji zachowania mechanicznego szkła akrylowego. Prowadzono badania mające na celu odniesienie mechaniki powstawania pęknięć obserwowanych przy teście Small Punch z wartościami wytrzymałości na pękanie zwykle stosowanymi w zestawieniu z dużymi próbkami w klasycznych metodach badawczych. Zastosowanie SPT jest szczególnie atrakcyjne, ponieważ może być wykorzystane do analizy zachowania sprężystego, jak i kruchego płaszcza cementowego wokół eksploatowanych urządzeń. Badania wskazały na znaczny wpływ temperatury tego materiału na wyniki, przy czym należy nadmienić wysoką dokładność odzwierciedlenia wpływu temperatury na wyniki w metodach SPT. Próbkki z cementu kostnego wykazały tendencję do plastycznego pęknięcia przy temperaturze ciała, a pęknięcia kruche już w temperaturze pokojowej. Praca odkształcenia była znacznie wyższa w temperaturze ludzkiego ciała.

Badanie SPT zostało wprowadzone do standardów badań dla biomateriałów UHMWPE (ang. Ultrahigh-molecular-weight polyethylene) (jak np. ASTM F2183), a wyniki badania [54] wskazują, że SPCT może być również z powodzeniem wykorzystywane do standardowych prób materiałowych na biomateriałach

PEEK (ang. Polyetheretherketone). Proces SPT jest bardzo czuły na zmiany parametrów, co widoczne jest w efektach, jakie są wartości maksymalnego obciążenia przed pęknięciem, na które wpływały zmiany w krystaliczności materiału, jego gradacji, procesach formowania i wyżarzania w produkcji. Podobnie siła wzrastała wraz z krystalicznością, a spadała wraz z dodatkiem materiałów kontrastujących. Zwiększenie krystaliczności materiałów zmniejszało również wartość przemieszczenia stempla przed zerwaniem próbki, prowadząc do mniejszego odkształcenia. Mimo że nie wykazano bezpośredniej zależności wyników do wielkoskalowych testów, takich jak próba jednoosiowego rozciągania, ze względu na wieloosiowość badań SPT, zaobserwowano wysoką czułość na zmiany wartości parametrów wynikowych już przy małej ilości materiału wymaganego do przeprowadzenia badania. Rokuje to na szerokie zastosowanie prób SPT do analiz materiałów PEEK w przyszłości. Co miało miejsce już w roku 2018, podjęto próbę analizy teoretycznej i eksperymentalnej modułów sprężystości polikaprolaktonowych materiałów PLC zmieszanych z organiczno-nieorganicznym wypełnieniem (PLC/TiO₂ oraz PLC/ZrO₂) [84]. Aktualnie badania prowadzi się również na czystym wolframie stosowanym w reaktorach fuzyjnych nowych konstrukcji, gdzie analizie poddano wartości temperatury przejścia w stan kruchy DBTT [126] oraz na superstopach Ni₃Al używanych do produkcji śmigieł turbin wiatrowych [24].

Kolejne lata przynosiły kolejne próby standaryzacji badania SPT dla polimerów, tak też w pracy [131] oraz wcześniej w 2013 [117] przeprowadzono analizę zdatności tej metody do określenia parametrów materiałowych filmów z nanokompozytów PLA (ang. Polylactide acid). W 2013 potwierdzono przy tym wysoką czułość testu SPT na obecność wad i zmian w objętości próbki materiału. Odpowiedni związek pomiędzy badaniem SPT a konwencjonalnym badaniem UTT został zaproponowany w formie równania (2.15) wyznaczającego granicę plastyczności σ_y i zawierającego w sobie stosunek P_y/t^2 będący stosunkiem obciążenia przy przejściu materiału w stan plastyczny do kwadratu grubości próbki przed testem.

$$\sigma_y = \frac{0.082P_y}{t^2} + 22.8 \quad (2.15)$$

Zastosowanie badania SPT do analizy polimerów było opracowywane w pracach [118, 155] gdzie dla różnych typów polimerów określono charakterystyki mechaniczne. Pod dwuosiowym stanem naprężenia charakterystycznym dla PST, polimery generalnie zachowują się bardziej plastycznie niż w standardowej próbie rozciągania, prowadząc do większych odkształceń przy zerwaniu. Region utraty stateczności polimeru można łatwo zidentyfikować w próbie SPT na wykresie po przyjsciu z wstępnego regionu sprężystego. Jest to dobrze widoczne przy materiałach nieumacniających się wraz z odkształceniem, jednak i te, które się umacniają, wykazują widoczny punkt przejścia w stan plastyczny. W czasie tego eksperymentu zidentyfikowano silne związki pomiędzy granicą plastyczności i modułem sprężystości z badania UTT w zestawieniu z charakterystyką krzywej SPT i jej wynikami. Nie wykazano jednak zależności pomiędzy granicą wytrzymałości a wydłużeniem przy zerwaniu próbki.

Badać można również wpływ powłok ochronnych takich jak naniesiona metodą HVOF (ang. High velocity oxygen coating) powłokę CoNiCrAlY o grubości około 400 μm [53]. Badania wskazały, że temperatura przejścia w stan kruchy BDTT dla takiej powłoki wyniosła około 500-750°C. Badanie SPT pozwoliło na wyznaczenie modułu sprężystości i obliczenie granicy plastyczności dla szerokiego zakresu temperatur. Moduł sprężystości spadł z około 237 GPa przy 500°C do 60 GPa przy 700°C, a granica plastyczności spadła

z 1300 MPa do zakresu 200-350 MPa. W temperaturze pokojowej obserwowane są kruche pęknięcia analizowanej powłoki ochronnej i propagacja pęknięć pomiędzy indywidualnymi ziarnami fazy β . Gdzie przy temperaturze 700°C dominuje plastyczne pękanie. Zakres ten został określony na podstawie wspomnianych różnic parametrów oraz po obserwacji zniszczonych próbek, wskazujących, w zależności od temperatury badania, na kruche pęknięcia gwieździste oraz krzyżkowe, lub też plastyczne pęknięcia szczelinowe pod czaszą.

Alternatywne podejście przyjęto do próby indywidualnej badania związku wyników SPT z klasycznymi próbkami dla stopów miedzi [37]. Stopy miedzi mają szerokie zastosowanie w elektrowniach jądrowych, gdzie wymagane są testy komponentów podczas ciągłego użytkowania. Wybrano czternaście stopów miedzi, dla których znane były własności mechaniczne. W badaniu zastosowano modele elementów skończonych dla metody SPT dla wszystkich stopów, a następnie obliczano i wyznaczano wartości obciążenia przy przejściu materiału w stan plastyczny P_y oraz maksymalnego obciążenia przed zerwaniem próbki P_{max} odpowiednio do granicy plastyczności i wytrzymałości. Jako zależności zaproponowano wzory (2.16) i (2.17):

$$\sigma_{YS} = 0.3586 \frac{P_y}{t_0^2} \quad (R^2 = 0.9417) \quad (2.16)$$

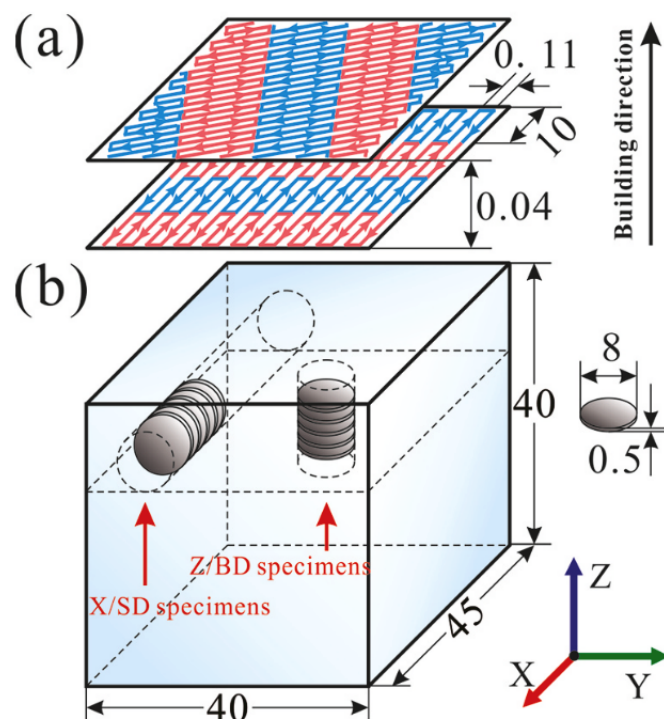
$$\sigma_{UTS} = 0.11315 \frac{P_{max}}{t_0^2} - 52.435 \quad (R^2 = 0.6979) \quad (2.17)$$

Gdzie σ_{YS} i σ_{UTS} to odpowiednio naprężenia równe granicy plastyczności materiału i granicy wytrzymałości, a t_0 to grubość początkowa próbki. Autorzy podają, że nowo wprowadzone zależności będą użyteczne przy projektowaniu nowych komponentów ze stopów miedzi, ponieważ SPT pozwala na określanie stanu materiału w trakcie użytkowania analizowanych komponentów.

Kolejne zastosowania metody SPT przedstawiono w pracy [67], kiedy to podjęto próbę ewaluacji SPT do przewidywania żywotności kabli z PVC. Oceniono degradację żywicy PVC podczas pracy z wysokimi temperaturami. Energie aktywacji oraz przewidywana żywotność komponentu były najniższe przy wykorzystaniu metody SPT ze wszystkich analizowanych metod do zastosowania przy badaniu tego materiału. Wskazuje to na czułość metody SPT na zachodzące w materiale zmiany mikrostrukturalne oraz wszelkie degradacje jego własności mechanicznych. Jak wykazano, jest to odpowiednia metoda do zastosowania przy analizie żywotności materiałów opartych na żywicach.

Kolejnym zastosowaniem SPT jest analiza materiałów wykonanych za pomocą metalurgią proszków. Takie badanie opisano w pracy [83] skupiono się na orientacji próbki w objętości materiału spiekane. Analizie anizotropii własności poddano superstop GH4169, wyprodukowane metodą SLM (ang. Selective Laser Melting), z którego pobrano próbki w dwóch orientacjach Z oraz X 2.11.

Po tej samej obróbce cieplnej i w tych samych warunkach obciążania próbka Z wykazała większą żywotność i plastyczność w teście SPCT niż próbka X ze względu na dynamiczną rekrytalizację zachodzącą w próbce Z w trakcie całego badania. Większa plastyczność była efektem heterogenicznej struktury ziaren, złożonych z niewielkiej objętości ziaren kolumnowych, zorientowanych w płaszczyźnie krystalograficznej $\langle 001 \rangle$, i dużej objętości frakcji ziaren V-kształtnych, zorientowanych w płaszczyźnie $\langle 011 \rangle$. W porównaniu z próbka X, próbka Z uzyskała większy potencjał energetyczny do osiągnięcia poziomu aktywacji rekrytalizacji.



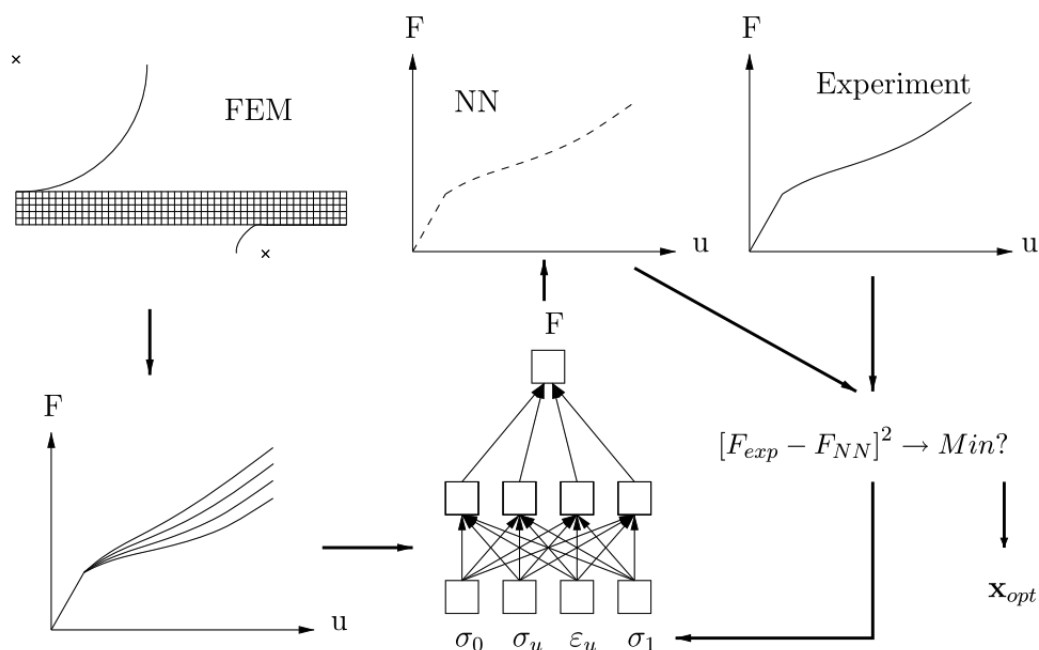
Rysunek 2.11: Schematyczna ilustracja a) strategii skaningowej z rotacją 67° pomiędzy warstwami; b) orientacji i wymiarów próbki SPCT pobranej z bloku materiału GH4169. [83]

Materiał niemetaliczny oraz wytwarzany alternatywnymi metodami wciąż stanowią wyzwanie dla skutecznego wykorzystania metod SPT do oceny ich stanu oraz odpowiedniej ekstrakcji wyników. Stanowi to potencjalne pole do dalszych badań w laboratoriach SPT na całym świecie, aby wzmocnić otrzymywane zależności i dopracować sposoby zastosowania praktycznego SPT.

2.8 Sieci neuronowe (NN) i uczenie maszynowe

Korelacja wyników badań metody SPT z metodami klasycznymi jest zagadnieniem, które stanowiło wyzwanie od początku powstania tej metody. Problem ten wynika z różnego stanu naprężeń w teście SPT i klasycznych testach własności materiałowych. Próby transformacji otrzymanych parametrów na ich odpowiedniki pozyskiwane ze standardowych prób z powodzeniem prowadzono dla konkretnych materiałów, jednak te same zależności nie zawsze były możliwe do uogólnienia na inne materiały. Nawet teraz unormowane sposoby obliczania danej własności nie są równie dokładne przy każdym użyciu dla różnych materiałów i wciąż wymagają wielu prób oraz kalibracji, aby osiągnąć zadowalający efekt w skali pozwalającej na wykorzystanie badania w przemyśle i podanie miarodajnych wyników. Aby przyspieszyć, a jednocześnie ograniczyć wpływ problematycznych przeliczeń zależnych od wielu parametrów, wysiłki badawcze skierowano w innym kierunku. W pracy [2] podjęto próbę wyznaczania parametrów materiałowych z wykorzystaniem sieci neuronowych (NN, z ang. Neural Networks). Sieci neuronowe były „uczone” przebiegu krzywych SPT. Krzywe te otrzymano z wykorzystaniem modeli materiałowych GTN. Nazwa pochodzi od nazwisk autorów Gurson, Tvergaard i Needleman, którzy opracowali teorię ciągłego sprężysto-plastycznego uszkodzenia

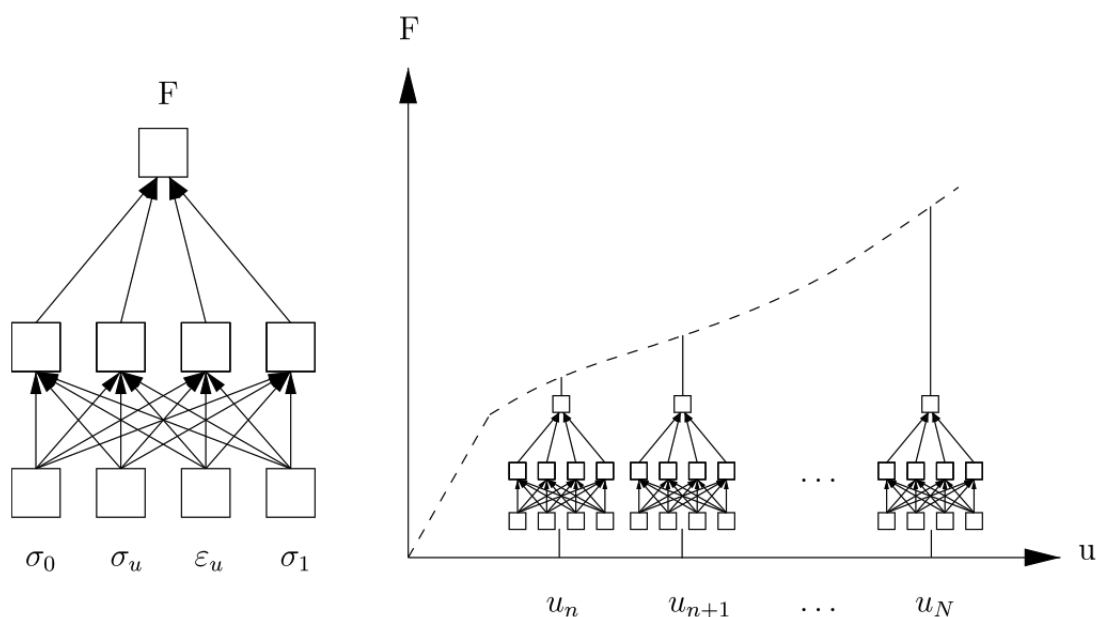
[40, 139]. W czasie procesu uczenia sieci neuronowe generują aproksymacje funkcji dla odwrotnego problemu, wiążąc parametry materiałowe z kształtem krzywej obciążenie – ugięcie. W wspomnianej pracy przeprowadzono badania dla trzech plastycznych stali (10MnMoNi55, 18Ch2MFA i StE690). Przed uczeniem i identyfikacją parametrów należy dokonać ich wstępnego wyboru, gdyż wykorzystanie wszystkich parametrów może być zbyt złożone i czasochłonne do przeprowadzenia. W rezultacie, w pracy [2], otrzymano zadowalające wyniki i pozytywną ocenę odtworzonych krzywych SPT z parametrami materiałowymi wszystkich trzech stali.



Rysunek 2.12: Identyfikacja parametrów materiałowych z użyciem sieci neuronowych. [78]

Sieci neuronowe mogą odwzorować skomplikowane zależności pomiędzy danymi wejściowymi i wyjściowymi. Dzięki ścieżce optymalizacji oraz nauczaniu z wykorzystaniem baz danych zawierających krzywe SPT oraz parametry materiałowe, mogą być one użyte do określania rzeczywistych krzywych naprężenie – odkształcenie (Rysunek 2.12). Pozwala to uniknąć stosowania czasochłonnych symulacji metodami elementów skończonych. Takie zastosowanie przyjęto z użyciem Stuttgart Neural Network Simulator (SNNS) w wersji 4.2 w pracy [78]. Aby uprościć symulacje z użyciem sieci neuronowych, pojedyncze sieci były generowane dla konkretnych przemieszczeń u_n aby zredukować ilość danych wejściowych do czterech parametrów materiałowych. Wszystkie 20 sieci o krokach $\Delta u = 0.1$ mm ustanowiono do aproksymacji całej krzywej obciążenie – ugięcie dla metody SPT. Struktura jednokierunkowej sieci neuronowej jest widoczna na rysunku (Rysunek 2.13). Każda z sieci neuronowych składa się z neuronów z sigmoidalnymi funkcjami aktywacji na wejściu (4 neurony), ukrytymi neuronami (także 4) oraz warstwą wyjściową (1 neuron). Neurony pierwszej warstwy otrzymują zewnętrzne wartości wejściowe (znormalizowane parametry materiałowe). Ich wartości ważone są wysyłane dalej do warstwy ukrytej, która podaje informacje do warstwy wyjściowej, która ostatecznie wydaje dane wyjściowe jako znormalizowaną siłę dla danego przemieszczenia u_n [78]. Nauczanie sieci neuronowych składa się ze zmieniania wag połączeń między kolejnymi warstwami, aby zminimalizować błędy pomiędzy odpowiedzią sieci i docelowymi wartościami po-

dawanymi przez wygenerowane w symulacjach lub otrzymane na drodze eksperymentów. Dane wejściowe są wybierane losowo, a sieci neuronowe są nauczone przez schemat Rprop (ang. Resilient backpropagation). Nauczanie uważa się za zakończone, gdy względny błąd wyników wynosi mniej niż 0,01, aby uniknąć przetrenowania. Po takim cyklu nauczania każda z sieci powinna być w stanie zwrócić znormalizowane wartości siły jako funkcję zbioru parametrów. Umiejętność generalizowania powinna być walidowana z niezależnymi wzorcami poza procesem nauczania. W ramach eksperymentu, w zestawieniu z wynikami prób rozciągania, krzywe wskazały na charakterystykę zależności od temperatury, gdzie wraz ze spadkiem temperatury krzywe przesuwają się na wyższe poziomy, gdzie obserwowane są wyższe siły przy takich samych przemieszczeniach. Jednocześnie materiał zachowuje się mniej plastycznie, co prowadzi do zmniejszenia maksymalnego przemieszczenia przed pęknięciem. Zmierzone krzywe podzielono na trzy grupy i sklasyfikowano je wizualnie według typu pęknięcia na przełomie.



Rysunek 2.13: Po lewej: struktura sieci neuronowej. Po prawej: każda sieć jest nauczana na danym przemieszczeniu. [78]

Sieci neuronowe są często trenowane z wykorzystaniem wyników symulacji testów SPT metodą elementów skończonych. Każda próbka treningowa to para wektora wejściowego, zawierającego przemieszczenie, oraz zestawu parametrów materiałowych jako wektor wyjściowy, np. obciążenie F . Jedna symulacja dostarcza krzywą, która następnie jest rozkładana na 100 próbek, gdzie u oraz F są zmiennymi [3]. Aby pokryć powierzchnię danego parametru, wykonuje się więcej symulacji ze zmiennymi parametrami. Może to być pomocne, jeśli zmienne parametry są dobierane losowo z określonej puli. Całkowita ilość potrzebnych próbek zależy od złożoności aproksymowanej funkcji oraz pożądanej dokładności. Do identyfikacji 4 parametrów hartowania wykorzystano tysiąc symulacji, a do identyfikacji pięciu parametrów uszkodzenia pięć tysięcy symulacji. Liczba niezbędnych neuronów w warstwach środkowych zależy również od złożoności problemu. Tak też dla parametrów utwardzania użyto dwudziestu pięciu neuronów, a dla parametrów uszkodzenia aż pięćdziesięciu. Do uczenia się sieci neuronowych próbki są dzielone na 2 zestawy. Do treningu wykorzystywany jest jeden zestaw zawierający 90% wszystkich próbek. Pozostałe wykorzystuje się tylko

do testowania, a nie do treningu. Podczas nauczania monitorowana jest dokładność zestawów uczących i walidacyjnych. Nauczanie jest przerywane, jeśli błąd walidacji osiągnie zakładaną, wcześniej przyjętą wartość, lub w przypadku stosowania mechanizmów kontroli uczenia, gdy nie następuje poprawa po określonej liczbie cykli uczenia, zwanych epokami. Kolejne prace potwierdzają wymóg ciągłego poszerzania i rozbudowywania bazy danych, gdzie autorzy analizy z pracy [134] uważają, że 100 zestawów danych to wciąż za mało, mimo ciągłego udoskonalania sieci neuronowej. Dopiero po kontynuacji eksperymentów i poszerzeniu bazy danych do ponad 300 rekordów uzyskano zadowalające wyniki. Średni błąd wyników dla granicy wytrzymałości wahał się między 1 a 3%, a dla granicy plastyczności wynosił 4-8%. Analizowane materiały to stale 10GN2MFA, 08Ch18N10T oraz 14MoV6-3, wykorzystywane w przemyśle energetycznym. W literaturze można znaleźć informacje o wykorzystaniu 457 grup hipotetycznych krzywych materiałowych w celach nauczania sieci. W tym wypadku odbyło się to z wykorzystaniem sieci neuronowej wstecznej propagacji BP (ang. Backpropagation) do generowania hipotetycznych krzywych naprężenie-odkształcenie [133]. W niniejszej pracy wykorzystano wyłącznie zasymulowane krzywe z obu testów SPT i UTT dla jednego rodzaju materiału w celu uczenia sieci neuronowej. Symulacje bazowały na modelu zniszczeniowym GTN (Surson-Tvergaard-Needleman) aby wygenerować dane wejściowe do nauczania. Na podstawie tych danych przygotowano sieć nakierowaną na rozpoznawanie jednego typu materiału. W pracy [90] badacze za cel postawili porównanie wydajności różnych modeli sieci neuronowych w celu bezpośredniej, odwrotnej identyfikacji parametrów materiału polimerowego typu ABS:

- Perceptron wielowarstwowy (z ang. *Multilayer Perceptron*) MLP
- Sieć konwolucyjna (z ang. *Convolutional Neural Network*) CNN
- Bayesowska sieć neuronowa (z ang. *Bayesian Neural Network*) BNN

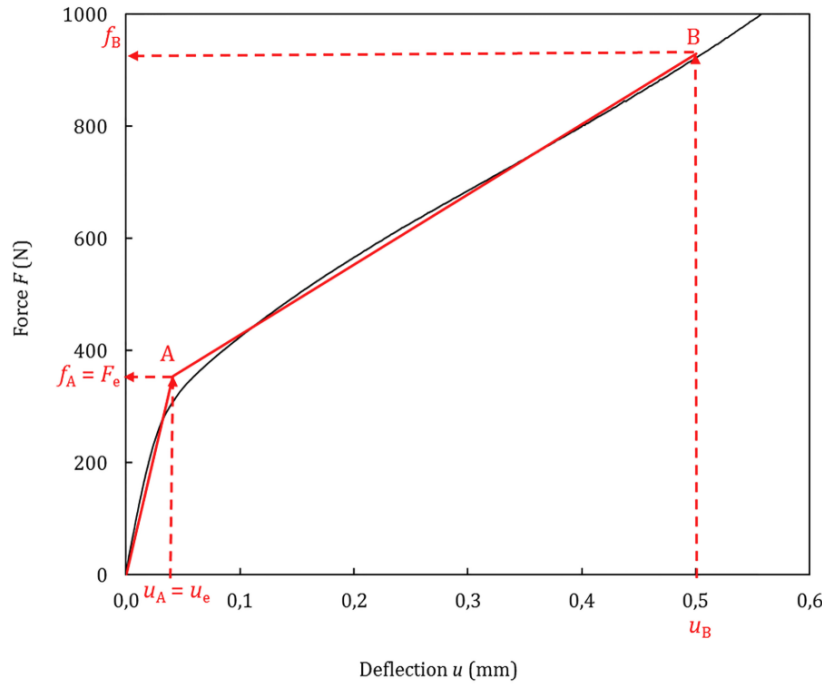
Jako dane wejściowe zastosowano wyłącznie zasymulowane, z wykorzystaniem MES, pomijając dane eksperymentalne, a modele sieci neuronowych przygotowano w środowisku Python. Wyniki pracy wskazują na wyższość sieci typu CNN pod kątem dokładności predykcji, jednak z ryzykiem przetrenowania sieci. Najnowsze badania zaczynają wykorzystywać rzeczywiste dane wejściowe z eksperymentów, jednak wciąż są wspierane w głównym nurcie przez symulacje MES [101]. Autorzy niniejszej pracy za cel obrali porównanie trzech metod uczenia maszynowego w celu bezpośredniej predykcji własności badanych stali:

- Metody propagacji wstecznej (z ang. *Back Propagation*) BP
- Sieci radialnych (z ang. *Radial Basis Function*) RBF
- Las losowy (z ang. *Random Forest*) RF

W wyniku analiz przedstawiono najwyższą skuteczność sieci typu BP pod kątem predykcji skalarnych parametrów badanych materiałów.

2.9 Standardowa metodologia analizy wyników testów SPT

Otrzymane w próbie SPT lub SPCT wykresy oraz dane odczytane z maszyny poddaje się dalszej analizie celem oceny stanu badanego materiału [148, 35, 37, 3, 44, 65, 30]. Na tej podstawie można wyznaczyć



Rysunek 2.14: Wyznaczanie wartości F_e oraz u_e metodą bi-linearną [18].

parametry takie jak granica plastyczności, granica wytrzymałości, wydłużenie próbki oraz odporność na pękanie [35], a także w przypadku próby SPCT odporność na pełzanie [65].

Do obliczania wartości siły F_e przy której nastąpiło przejście ze stanu sprężystego do plastycznego podczas próby SPT, wykorzystuje się model dwuliniowy, gdzie F_e jest uzyskane z wykresu poprzez przeprowadzenie funkcji dwuliniowej $f(u)$ (2.18) przez punkty A i B na wykresie siły w funkcji ugięcia (Rysunek 2.14). Aby zminimalizować błędy aproksymacji, stosuje się dodatkową funkcję err (2.19). Zalecane jest dobieranie wartości u_B równej grubości badanej próbki. Korzystając z wykresu, można również wyznaczyć maksymalną siłę potrzebną do zainicjowania pęknięcia próbki F_m oraz całkowitą energię pęknięcia (2.20) [18].

$$f(u) = \begin{cases} \frac{f_A}{u_A} & \text{dla } 0 \leq u < u_A \\ \frac{f_B - f_A}{u_B - u_A}(u - u_A) + f_A & \text{dla } u_A \leq u \leq u_B \end{cases} \quad (2.18)$$

$$err = \int_0^{u_B} [F(u) - f(u)]^2 du \quad (2.19)$$

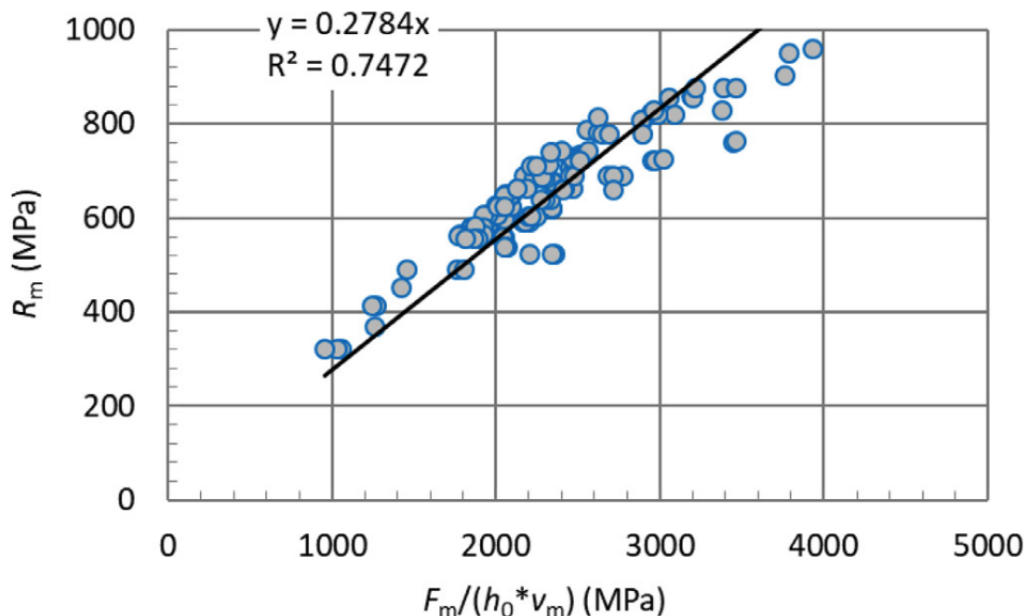
$$E_{sp} = \int_0^{u_B} F(u) du \quad (2.20)$$

W publikacjach [35, 44, 20, 9, 157] przedstawiono oraz podparto udanymi eksperymentami metody korelacji danych uzyskanych z prób SPT z własnościami mechanicznymi badanych materiałów. Tak też zaproponowano wykorzystanie maksymalnej siły użytej w teście F_m według wzoru (2.21) lub wartości siły w momencie utraty plastycznej stateczności materiału F_i (2.22) do wyznaczenia granicy wytrzymałości R_m badanego materiału. Metodyka ta została ujęta w normie EN 10371:2021 [18]. Do estymacji stosuje się

współczynnik β_{R_m} , którego wartość jest różna w zależności od przyjętej metody. Dla obliczeń wykorzystujących siłę maksymalną przyjmuje się jego wartość w przedziale 0,19-0,32, w zależności od badanego materiału i w zestawieniu z wynikami badań w klasycznych testach wytrzymałościowych. Dla przykładowych badań nad 9 różnymi stalami w różnych temperaturach otrzymano wartość współczynnika $\beta_{R_m} = 0.278$, jak przedstawiono na wykresie (Rysunek 2.15). W przypadku wykorzystania F_i wartość współczynnika β_{R_m} dobiera się z tabeli (2.3) adekwatnie do stosowanej geometrii próbki oraz zestawu testowego. Parametry te otrzymano z wykorzystaniem symulacji. Metoda wykorzystująca F_i została zweryfikowana dotychczas tylko na stalach ferrytyczno-martenzytycznych [18]. Obie metody można wykorzystywać zamiennie, zależnie od generowanych w próbach typów wykresów danych [35, 3, 43].

$$R_m = \beta_{R_m} \cdot \frac{F_m}{h_0 \cdot u_m} \quad (2.21)$$

$$R_m = \beta_{R_m} \cdot \frac{F_i}{h_0^2} \quad (2.22)$$



Rysunek 2.15: Korelacja pomiędzy wartością F_m uzyskaną w próbie SPT oraz znanymi wartościami granicy wytrzymałości R_m badanych materiałów. [9]

Φ [mm]	r [mm]	D [mm]	Typ krawędzi w matrycy dolnej	h_0 [mm]	Typ krzywej	$u_i v_i$ [mm]	β_{R_m}
8	1.25	4.0	Fazowana 0.2x45°	0.5	$F(u)$	0.552 (u_i)	0.192
8	1.25	4.0	Fazowana 0.2x45°	0.5	$F(v)$	0.645 (v_i)	0.179
3	0.5	1.75	Fazowana 0.2x45°	0.25	$F(u)$	0.282 (u_i)	0.205
3	0.5	1.75	Fazowana 0.2x45°	0.25	$F(v)$	0.320 (v_i)	0.197

Tabela 2.3: Zależność współczynnika β_{R_m} od geometrii próbki i zestawu testowego. [18]

Również w publikacjach [35, 42], gdzie w analizie uwzględniono kilka modeli umocnienia odkształceniowego oraz w normie EN 10371:2021 [18] zawarto wyniki analiz korelacji danych otrzymanych w próbie

SPT z granicą plastyczności. Granicę plastyczności $R_{p0.2}$ estymować można z wykorzystaniem wartości siły F_e w momencie przejścia materiału ze stanu sprężystego w plastyczny według wzoru (2.23). Wartość współczynnika $R_{p0.2}$ dobierana jest z tabeli (2.4). Wartości te zostały wyznaczone dla stali z granicami plastyczności w przedziale 200-1000 MPa.

$$R_{p0.2} = \beta_{R_{p0.2}} \cdot \frac{F_e}{h_0^2} \quad (2.23)$$

Φ [mm]	r (mm)	D (mm)	Typ krawędzi w matrycy dolnej	h_0 (mm)	Typ krzywej	$\beta_{R_{p0.2}}$
8	1.25	4.0	Fazowanie 0.2 x 45°	0.5	$F(u)$	0.510
8	1.25	4.0	Fazowanie 0.2 x 45°	0.5	$F(v)$	0.479

Tabela 2.4: Proponowane wartości współczynnika $\beta_{R_{p0.2}}$ dla standardowej geometrii zestawu testowego i danych parametrów próbki. [18]

Na podstawie badań SPT możliwa jest estymacja temperatury przejścia materiału ze stanu plastycznego w stan kruchy [18, 20, 7, 154]. Jest to niezwykle istotna informacja przy projektowaniu maszyn pracujących w środowiskach morskich lub w regionach z okresową bądź stałą niską temperaturą pracy. W publikacji [41] zaprezentowano stałą korelację między temperaturą przejścia w stan kruchy T_{SP} uzyskaną w testach Small Punch a wartością uzyskaną w próbie Charpy'ego T_{CVN} , którą można przedstawić za pomocą wzoru (2.24), gdzie wartość α wynosi około 0,42 dla próbek o standardowej średnicy.

$$T_{SP} = \alpha \cdot T_{CVN} \quad (2.24)$$

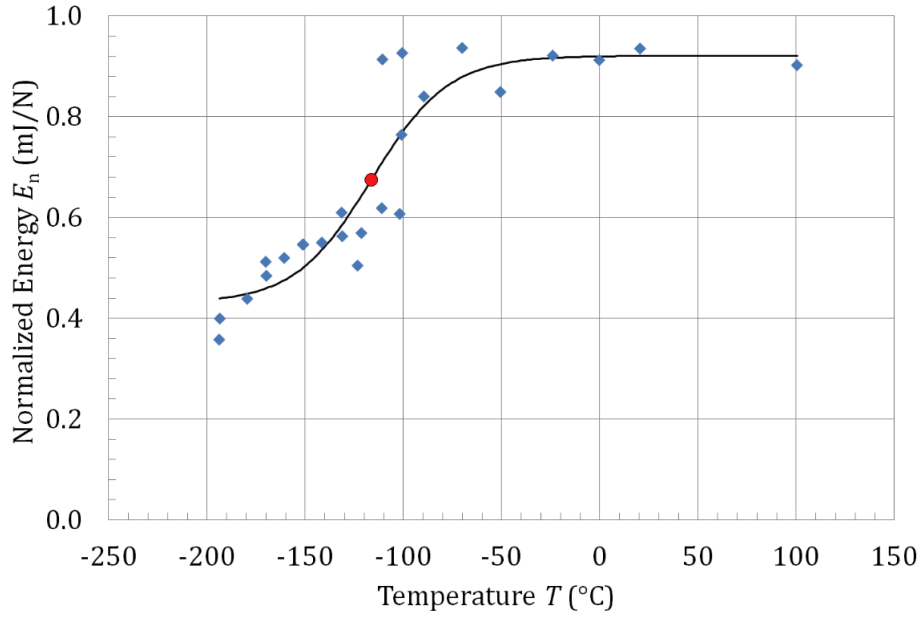
Jedynie przeprowadzenie wielu testów (minimum 12) w różnych temperaturach pozwoli określić dokładną wartość temperatury przejścia w stan kruchy. Można tego dokonać poprzez znormalizowanie całkowitej energii E_n użytej w próbie ESP według wzoru (2.25) oraz dopasowanie krzywej do otrzymanych wartości z zastosowaniem wzoru (2.26). Należy zastosować metodę najmniejszych kwadratów w celu wyznaczenia współczynników A, B, C oraz wartości T_{SP} zgodnie ze wzorem (2.27) [18]. W efekcie czego można sformułować krzywą (Rysunek 2.16), z której odczytuje się właściwą wartość T_{SP} . Jednostka temperatury może być podawana zarówno w K, jak i w °C.

$$E_n = \frac{E_{SP}}{F_m} \quad (2.25)$$

$$E_n(T) = A + B \cdot \tanh \left[\frac{T - T_{SP}}{c} \right] = \frac{E_{US} + E_{LS}}{2} + \frac{E_{US} - E_{LS}}{2} \cdot \tanh \left[\frac{T - T_{SP}}{c} \right] \quad (2.26)$$

Gdzie E_{US} są wartościami energii całkowitej z prób powyżej przewidywanej temperatury przejścia w stan kruchy, a E_{LS} poniżej tej wartości.

$$\sum_{i=1}^N \left[A + B \cdot \tanh \left(\frac{T_i - T_{SP}}{c} \right) - E_n(T_i) \right]^2 \rightarrow \min \quad (2.27)$$



Rysunek 2.16: Przykładowa krzywa wyznaczająca wartość T_{SP} z użyciem znormalizowanej energii dla stali P92. Wynikowa wartość $T_{SP} = -116$ °C zaznaczona na wykresie.

Alternatywnie w analogiczny sposób, możliwe jest wykorzystanie efektywnego odkształcenia pęknięcia ε_f (2.28) zmierzonego przy pęknięciu próbki, gdzie wartość znormalizowanej energii E_n zastąpiona jest wartością odkształcenia. Gdzie h_f jest grubością próbki zmierzona przy pęknięciu.

$$\varepsilon_f = \ln \left(\frac{h_0}{h_f} \right) \quad (2.28)$$

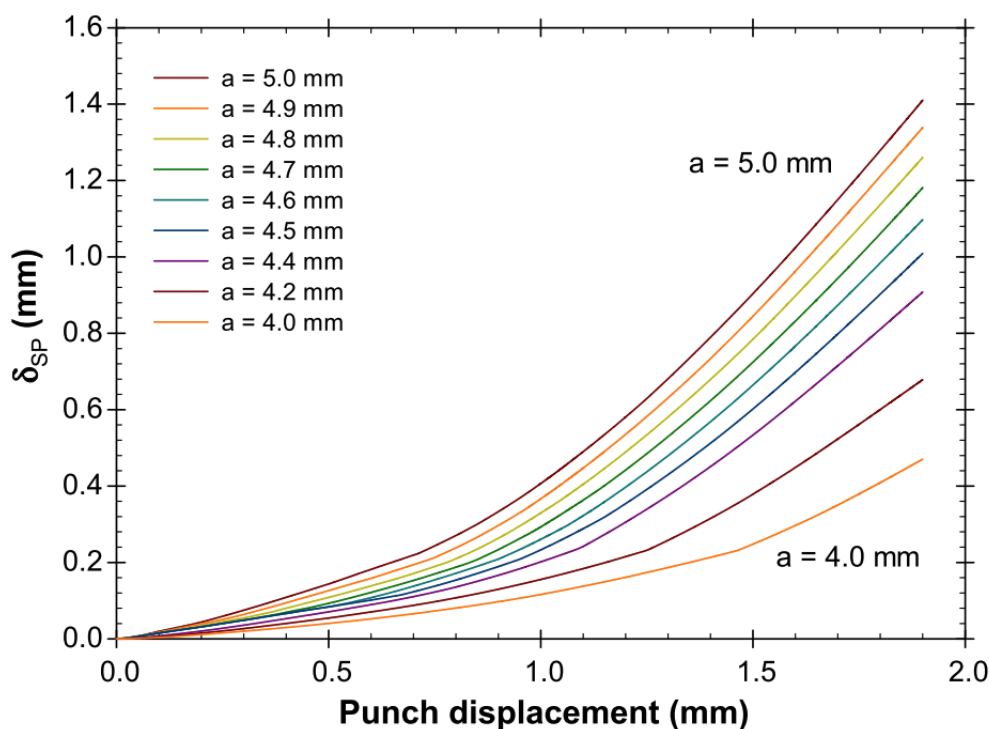
Materiały badane metodami SPT mogą być poddane ocenie odporności na pęknięcie. Odporność tę można definiować na trzy sposoby. Poprzez korelację T_{SP} z naprężeniem K_{IC} [MPa/m] (2.29) [127], zależną znacząco od grubości badanego materiału. Poprzez podejście energetyczne łączące ε_f z całkowitym kryterium wyznaczania energii pęknięcia J_{IC} [N/mm] (2.30) gdzie współczynniki k oraz J_0 są zależne od badanego materiału i można je znaleźć w literaturze [35]. Przy tym podejściu materiały plastyczne wykazują mniejszy wpływ grubości próbki. Ostatnią z głównych trzech metod określania odporności na pęknięcie w próbach SPT jest podejście czysto geometryczne z użyciem nacinanych próbek [18, 5].

$$K_{IC} = \frac{6600}{60 - (T - T_{CVN})} \quad (2.29)$$

$$J_{IC} = k \cdot \varepsilon_f - J_0 \quad (2.30)$$

Poprzez użycie krzywych (Rysunek 2.17) wyznaczonych poprzez wykonanie prób SPT na naciętych próbkach, możliwe jest przedstawienie geometrycznej metody definiowania odporności na pęknięcie z użyciem J_{IC} (2.31) gdzie δ_{IC} jest wielkością propagacji pęknięcia lub też wielkością odsunięcia początku nacięcia od pierwotnego położenia [18, 5].

$$J_{IC} = R_{p0.2} \cdot \delta_{IC} \quad (2.31)$$



Rysunek 2.17: Zależność wstępnej wielkości nacięcia od przemieszczenia stempla w próbie SPT. [5]

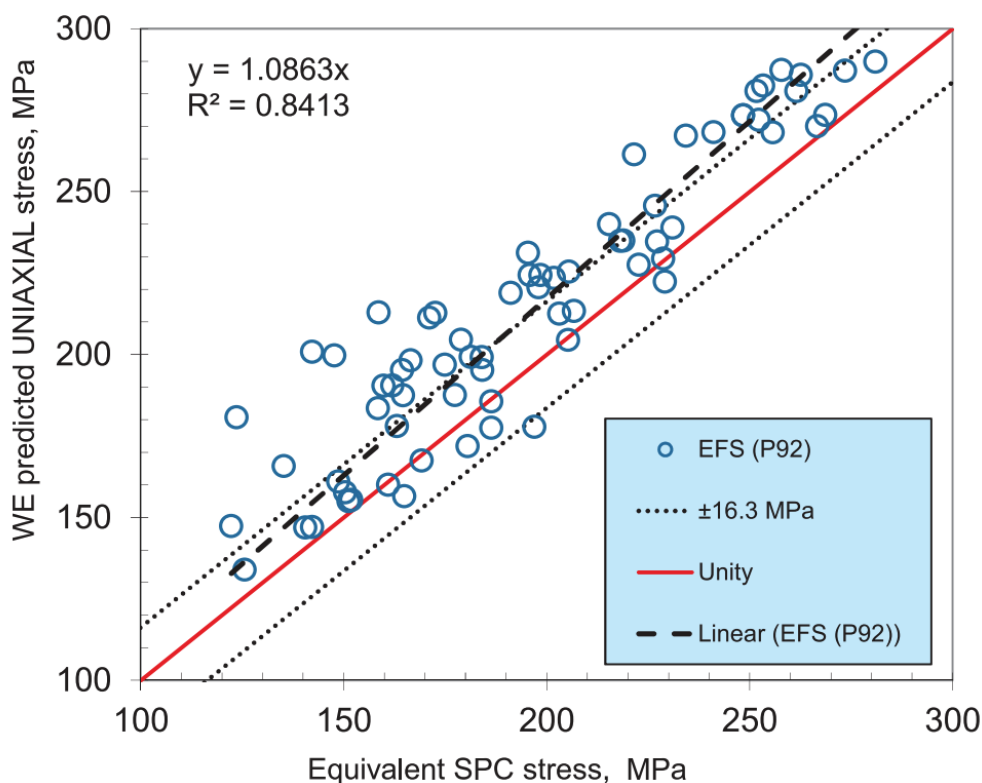
Ostatnią możliwą do określenia w próbach SPT własnością badanego materiału jest jego odporność na pełzanie [148, 43, 149, 29]. Próby pełzania wykonuje się w teście SPCT. Estymacja własności pełzania materiałów badanych tą metodą jest problematyczna ze względu na błędy i niejasności wynikające z przełożenia zadawanej w próbie siły F na równoważny jednoosiowy stan naprężenia σ . Przybliżenia pozwalają przyjąć wartość współczynnika Chakrabarty'ego $\Psi = 1,890$ N/MPa dla standardowej geometrii próbki SPT. Współczynnik ten jest stosunkiem wspomnianej wyżej siły F i jednoosiowego naprężenia σ dla takiego samego czasu do pęknięcia próbki w metodzie SPCT i klasycznej metodzie badania pełzania. Optymalizacje i przybliżenia zostały ustanowione na podstawie wielu źródeł [43]. Współczynnik ten może być dalej optymalizowany z wykorzystaniem wzoru (2.32). Gdzie dodatkowym czynnikiem korygującym jest minimalna prędkość pełzania u odnotowana w trakcie testu i podawana w [mm/h] (2.32) [18].

$$\Psi = \frac{F}{\sigma} = 1.916 u_{\min}^{0.6579} \left(\frac{N}{\text{MPa}} \right) \quad (2.32)$$

Dalsze optymalizacje zależności między klasycznym jednoosiowym testem pełzania materiałów a metodyką SPCT wymagają pozyskania większej ilości danych z ośrodków krajowych i zagranicznych. Na przykładzie (Rysunek 2.18) przedstawiono korelację interpolacji wyników badań jednoosiowego pełzania przeprowadzonych na stali P92 według modelu Wilshire'a [152] oraz równorzędnego naprężenia przy teście SPCT wyznaczonego według wzoru (2.32) [43].

Możliwe jest również wyznaczenie wartości minimalnej prędkości odkształcenia $\dot{\epsilon}_{\min}$ [1/h] (2.33) korzystając z analogicznej metody jak w przypadku równorzędnej wartości naprężenia pełzania.

$$\dot{\epsilon}_{\min} = 0.3922 \cdot \dot{u}_{\min}^{1.1907} \quad (2.33)$$



Rysunek 2.18: Przykładowe zestawienie wartości naprężenia pełzania dla stali P92 w teście SPCT z wartościami z klasycznej próby pełzania interpolowanymi modelem Wilshire'a [43, 152]

2.10 Aktualne trendy w rozwoju metody SPT

W roku 2023, w Santander w Hiszpanii odbyła się konferencja Small Sample Testing Techniques (SSTT) 2023. W konferencji udział wzięli badacze z całego świata zajmujący się zagadnieniem testów mechanicznych na niewielkich próbkach, do których należy SPT. Na konferencji prezentowano prace i badania w następujących tematach.

2.10.1 SPT na zaawansowanych materiałach

W pracy *Use of Small Punch Test (SPT) in the fatigue characterization of S355 structural steel* [100] zaproponowano z powodzeniem zastosowanie metody SPT do charakteryzacji własności stali strukturalnej S355 poprzez poddanie próbek cyklicznym obciążeniom aż do uzyskania uszkodzenia.

W pracy *J-R curve evaluation using Small Punch Test and multiple specimen technique* [6] – celem było udoskonalenie metody określania odporności na pękanie dla dwóch stali: AISI 4130M oraz AISI 4137. Metodologia stopniowego zwiększania wielkości nacięcia próbki SPT udało się jedynie dla pierwszej stali. W pracy *Assessing the Mechanical Behavior of an Additive Manufactured Nickel Based Superalloy Using Alternative Small Scale Test Techniques* [72] badania przeprowadzono na materiale LPBF IN718 w dwóch wariantach: próbki z kutego materiału oraz z wytworzonego metodami przyrostowymi. Wyniki porównania pomiędzy różnymi metodami, porównując je z metodami klasycznymi. Dla badań na miniaturowych prób-

kach znaczącym czynnikiem była wielkość uzyskanego ziarna. Badanie Shear Punch wskazało nieco lepszą korelację niż SPT względem klasycznych metod.

Determination of Fracture Toughness of Metallic Materials by Small Punch Test with a new type of slot structure [145] – autorzy badali wpływ geometrii nacięcia na próbce SPT na dokładność korelacji wyników odporności na pękanie. Przetestowano odmienne do klasycznego nacięcia rowki po grubości próbki oraz dodatek niewielkiej szczeliny przelotowej na środku próbek na ich zachowanie podczas badania.

Proficiency testing of material parameters assessment for SPT methodology precision increase [68] – badacze z instytutu UJV w Republice Czeskiej przeprowadzili serię testów w kilku ośrodkach badawczych na zasadzie round-robin celem porównania wyników otrzymywanych metodami SPT pomiędzy ośrodkami. Dodatkowo podjęto próbę udoskonalenia korelacji SPT oraz UTT. Szansą dla AGH jest udział w kolejnej rundzie tego typu badań, aby dołączyć aktywnie do środowiska instytutów zajmujących się metodami SPT.

Evaluation of material deformation in small punch fatigue with DIC technique [103] – autorzy z ośrodka badawczego w Indiach przedstawili swoją pracę nad wykorzystaniem metody Digital Image Correlation podczas badania SPT. Wyniki wspierane są symulacjami i nadal wymagają udoskonalenia. Obecnie największą przeszkodę stanowi wieloosiowość badania SPT i konieczność wykorzystania wielu kamer do dokładnego zmapowania odkształcanej powierzchni.

Overview of the study on fatigue response of an Inconel alloy through Cyclic Ball Indentation and Cyclic Small Punch Testing [104] – jeden z autorów kontynuuje badania przedstawione w poprzedniej prezentacji o wykorzystaniu metody DIC i skupia się na porównaniu metod do określenia wytrzymałości zmęczeniowej badanego materiału. Wyniki wskazały, że metoda Cyclic SPCT jest w stanie wskazać zadowalającą dokładnością odporność zmęczeniową dla Inconel 617. Specjaliści z tego ośrodka zaoferowali możliwość współpracy w dalszych badaniach.

2.10.2 Alternatywne zminiaturyzowane metody testowe

Mechatronics design and architecture of evaluation software for a portable instrumented indentation system PIIS3000™ [99] - Pan Daniel Omacht, który jest producentem sprzętu dla laboratorium SPT w Centrum Energetyki AGH, zaprezentował nowe urządzenie wykorzystujące cykliczne obciążanie powierzchni materiału bez jego permanentnego uszkodzenia, celem uzyskania jego parametrów.

In hot cell shear punch testing of neutron irradiated steels [31] – autorzy wykorzystali metodę Shear Punch do oceny stanu materiału, porównując wyniki z metodami klasycznymi. Badania przeprowadzono na napromieniowanych stalach w laboratorium w Los Alamos. Metoda Shear Punch nie została jeszcze dobrze zweryfikowana, jednak jest obiecującym dodatkiem do SPT.

2.10.3 SPCT

Application of small punch creep for creep residual life and life extension of gas turbine blades and vanes [142] – autorzy poczynili kroki w kierunku dokładniejszej korelacji wyników testów SPCT ze standardowymi badaniami pełzania.

Evaluation of texture strengthening effect of pure zirconium sheet by small punch test [136] – w niniejszym badaniu wykorzystano badanie SPT w celu ewaluacji właściwości czystego cyrkonu. Wsparto się

tutaj symulacjami komputerowymi, a analizowanym aspektem była anizotropowość właściwości materiału i umocnienie powierzchniowe. Otwiera to drogę do badań innych materiałów niż stopy stali.

Comments on standard EN-10371 and proposal for standard follow up [77] – Członkowie komitetu opracowującego normę oraz zrzeszającego ekspertów w dziedzinie SPT przedstawiali kierunek dalszych działań i plany, co należy wprowadzić w kolejnej rewizji wspomnianego standardu. Do głównych celów komitetu należy:

1. Opracowanie semi-empirycznej lub semi-analitycznej metody pozwalającej na lepszą korelację wyników ze standardowymi metodami.
2. Analiza pękania kruchego materiału i kształtu pęknięcia.
3. Utworzenie szeroko dostępnej bazy danych materiałowych

Correlation and extrapolation of small punch creep test data for Incoloy 800H and 347H heat-resistant alloys [57] – w tym badaniu autorzy przedstawili nową metodę korelacji siły do naprężenia pomiędzy badaniem Small Punch Creep Test a klasycznym badaniem pełzania. Dokładność korelacji zależy od oprzyrządowania i jego geometrii oraz testowanego materiału, jednak pozwala już na znacznie szersze wykorzystanie niż wzór korelacji obecnie znany w literaturze.

Investigation of deformation mechanism and creep behavior of 347H stainless steel in small punch creep test [58] – autorzy poprzedniego badania kontynuują analizę i ocenę metody SPTC, tym razem prowadząc badania na stali 347H oraz wspierając się analizą elementów skończonych.

Evaluation of Mechanical Yield Properties of 17-4PH using Small punch and sub-size Tensile Testing [150] – w niniejszym badaniu dokonano zestawienia i porównania wyników z badań SPT w dwojakim podejściu do przetwarzania wyników oraz metody jednoosiowego rozciągania w zminiaturyzowanej skali na specjalnie przystosowanym urządzeniu. Korelacje wykonano z powodzeniem, potwierdzając i przedstawiając dokładność wszystkich metod.

2.10.4 SPT w środowiskach agresywnych

Optimal steel times for ASTM F1624 standard methodology application to Small Punch Tests in aggressive environments [13] – w niniejszym badaniu dokonano dostosowania długości kroków czasowych podczas badań wykonywanych metodami stałych stopni obciążenia z wykorzystaniem SPT. Za bazę przyjęto normę ASTM F1624 wykorzystywaną do klasycznych testów.

Evolution of the Small Punch Test procedure to analyze the hydrogen embrittlement phenomenon [11] – grupa badawcza SIMUMECAMAT w przedstawionym artykule dokonała analizy zdatności metody SPT przy charakterystyce zjawiska, jakim jest kruchość wodorowa. Wytypowano najlepszą geometrię próbki SPT z nacięciem, aby zwiększyć powierzchnię wpływu wodoru.

Effect of post-weld treatment on the hydrogen embrittlement of CrMoV steel welded joints using Small Punch Test [79] – autorzy dokonali analizy możliwości i czułości metody SPT na badanie stali w obszarze spawu, strefy wpływu ciepła oraz bazowego materiału w stanie przed i po próbie cieplnej. Metoda SPT wykazała wystarczającą dokładność, aby z jej wykorzystaniem zaobserwować zjawisko kruchości wodorowej w każdym miejscu i stanie badanego materiału.

2.10.5 Numeryczne i empiryczne podejścia do badania SPT

Investigation on regression model for the force of small punch test using machine learning [156] – przedstawione badanie z wykorzystaniem nauczania maszynowego wskazało, że siła z badania SPT może być wyrażona jako binarny, liniowy model regresji granicy plastyczności i granicy wytrzymałości. Dokładność modelu została zweryfikowana na hipotetycznych materiałach oraz stali X70 i Incoloy 800H. Dalsze badania i udoskonalanie metody nadal są wymagane.

Characterisation of Shape Memory Alloys Using the Small Punch Test [1] – w niniejszym badaniu z powodzeniem udowodniono możliwość wykorzystania SPT do badania niektórych własności stopów żelaza z pamięcią kształtu. W tym celu wykorzystano eksperyment SPT wspierany modelem materiału z analizą elementów skończonych.

Enhancing understanding of hydrogen assisted fracture in small punch test through computational modelling [10] – w niniejszej pracy ponownie potwierdzono możliwość wykorzystania metody SPT do analizy wpływu wodoru na własności próbki. Badania wsparto symulacjami numerycznymi.

Room and elevated temperature Small Punch Static Testing and analysis using empirical and numerical methods [143] – autorzy wykorzystali analizę numeryczną w celu optymalizacji uzyskania parametrów materiałowych z metody SPT. Dodatkowo przedstawiono autorską metodę pozyskiwania materiału do badania SPT. Podobnie jak znana metoda scooping, ta również, przy właściwym użyciu, nie zaburza struktury badanej konstrukcji. Metoda ta pozwala na pobranie rdzenia materiału o średnicy 8 mm, gotowego do pocięcia do badania SPT, a następnie zaspawania powstałego otworu.

2.11 Podsumowanie przeglądu literatury

Mimo wieloletnich badań nad rozwojem metody, największy postęp zaobserwowano w ostatnich latach, kiedy sformułowano i opublikowano europejski standard EN 10371:2021 [18]. Przegląd literatury oraz przedstawienie aktualnych trendów światowych zaobserwowanych na konferencji tematycznej wskazują na rozwój zastosowań metody SPT, a nie na dalsze jej udoskonalanie. Otwarty jest kierunek automatyzacji oraz poprawy wyników otrzymywanych z metody SPT, znajdującej zastosowanie w praktycznych aplikacjach przemysłowych. Wskazuje to na otwarty, dalszy kierunek do poprawy jakości otrzymywanych z niej danych poprzez wprowadzenie wykorzystania sieci neuronowych, służących do przybliżania operatora do uzyskania wymaganego wyniku. Ogólne źródła, w tym badania opisane na przestrzeni ostatnich lat, prowadzone równolegle do niniejszej pracy, dotyczące wykorzystania nauczania maszynowego, najczęściej ograniczają się do prób regresji skalarnej poprzez predykcję bezpośrednich wartości granicy plastyczności i wytrzymałości oraz budowy modeli z wykorzystaniem symulacji komputerowych. Nie ujmują one realnego zastosowania praktycznego metod i ich implementacji, co pozwoliłoby na szerszy rozwój aplikowanych w przemyśle podejść. Opisywane modele są monolityczne, przyjmujące dane wejściowe w sposób jednolity dla każdego materiału lub poprzez dodanie sztucznych ograniczeń na potrzeby badań. Brak jest specjalizacji materiału do konkretnego stanu w opisywanych podejściach, zwłaszcza w przypadku analiz materiałów zmęczonych z pracujących konstrukcji na skraju żywotności. Co należy podkreślić, to że wszystkie przeanalizowane prace opisują wykorzystanie modeli bazujących na symulacjach komputerowych, z ustępstwami

dostosowanymi do idealnej sztucznej krzywej materiałowej. Niniejsza praca ma na celu przedstawienie i poszerzenie nowych podejść do zagadnienia automatyzacji predykcji stanu materiału, dzięki wykorzystaniu rzeczywistych danych z próbek pobieranych z materiałów eksploatowanych, bazując na wynikach eksperymentalnych oraz dostosowaniu modelu klasyfikacji do naturalnych niedoskonałości występujących w badanych materiałach.

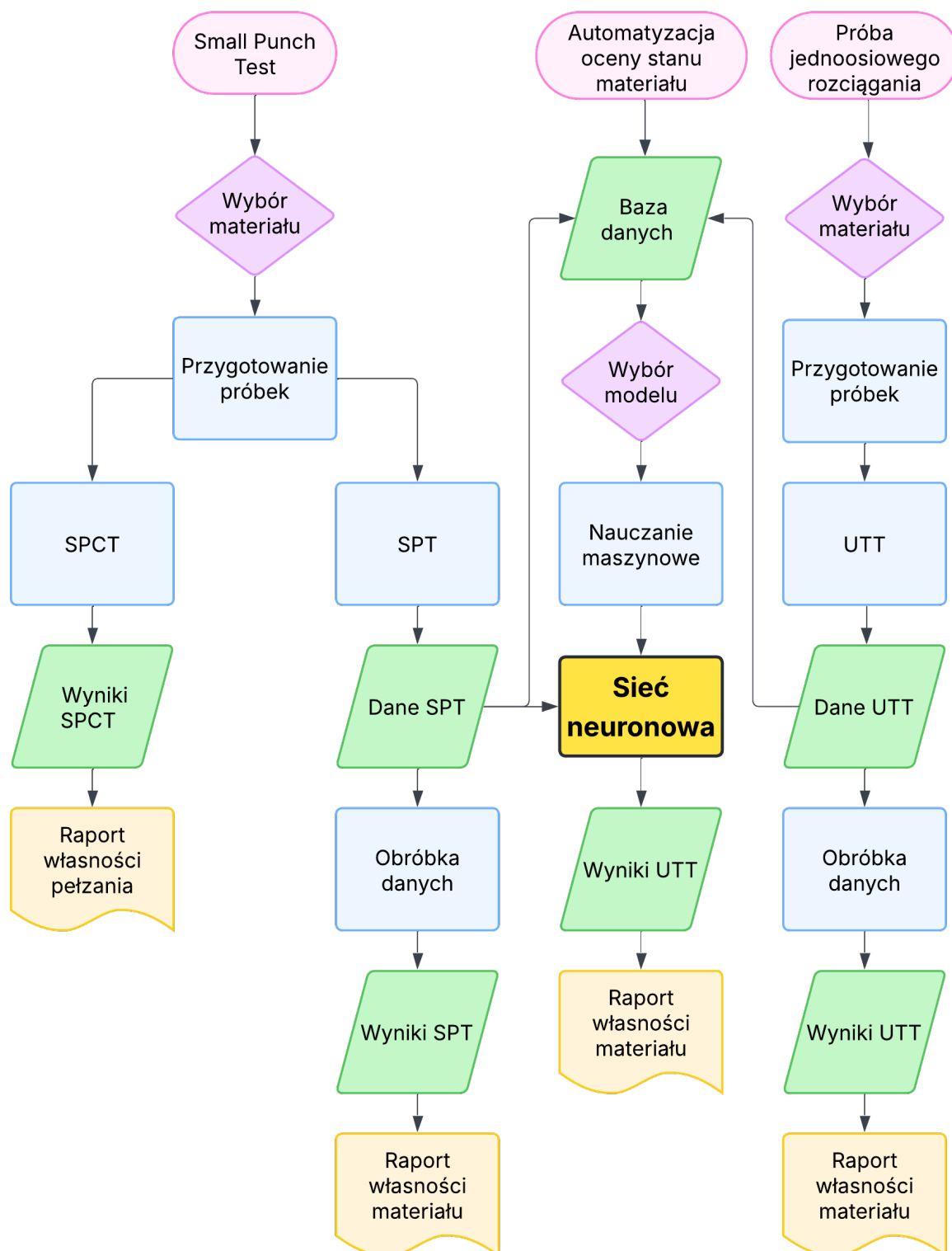
Rozdział 3

Metodologia badań

Ze względu na multidyscyplinarny charakter pracy jej metodologia może być podzielona na segmenty, do których należą: badania eksperymentalne, w ramach których realizowane były badania SPT oraz UTT, badania komputerowe, mające na celu realizację głównego zagadnienia badawczego pracy oraz metody obróbki, analizy i interpretacji wyników zarówno z badań eksperymentalnych, jak i komputerowych. Schemat blokowy przedstawiający ideę pracy został ujęty na rysunku 3.1. Przedstawia on podstawową ścieżkę postępowania przy ocenie własności materiału w przypadku badania SPT, jak i UTT. W obu przypadkach pierwszym krokiem po wyborze materiału do badania jest przygotowanie próbek, następnie wykonanie samego badania w celu pozyskania danych w formie surowej, które należy za pomocą dodatkowego oprogramowania lub narzędzi pomocniczych przetworzyć do formy pozwalającej na interpretację otrzymanych wyników. W ramach pracy powstała trzecia ścieżka, równoległa do obu badań, która w realizacji tworzy pomost pomiędzy badaniem SPT i UTT. Na podstawie wyników z obu badań ujętych w bazie danych przygotowano model sieci neuronowej z wykorzystaniem nauczania maszynowego, a następnie zaimplementowano go do całościowej sieci neuronowej. Sieć przyjmuje dane SPT, przetwarza ich sygnały do formy gotowej do implementacji do modelu sieci, przetwarza dane, a następnie na podstawie wyników odtwarza dane z badania UTT, jak została nauczona. Wyniki UTT wygenerowane przez sieć neuronową zostają dalej przedstawione na wykresach naprężenie-odkształcenie, a z nich odczytywane są własności materiału w postaci granicy plastyczności i granicy wytrzymałości. Wyniki dostępne są w każdej formie dla operatora do samodzielnej obróbki oraz interpretacji.

3.1 Materiały wybrane do badań

W ramach wszystkich prac nad automatyzacją oceny stanu materiału oraz badań wstępnych wykorzystywano stale kotłowe i rurociągowe. Stale kotłowe i rurociągowe stanowią grupę materiałów metalowych, które są projektowane z myślą o pracy w warunkach wysokiej temperatury, ciśnienia oraz w środowiskach narażonych na korozję. Wyróżniają się one właściwościami mechanicznymi, takimi jak wysoka wytrzymałość na rozciąganie, odporność na pełzanie oraz dobra plastyczność w podwyższonych temperaturach. Są to stale niskostopowe lub stopowe, których skład chemiczny został zoptymalizowany w celu zwiększenia



Rysunek 3.1: Schemat blokowy do zagadnienia automatyzacji oceny stanu materiału

odporności na działanie pary wodnej, gazów przemysłowych oraz czynników chemicznych obecnych w instalacjach energetycznych i chemicznych.

Dzięki dodatkom stopowym, takim jak chrom, molibden, wanad czy nikiel, stale te charakteryzują się zwiększoną odpornością na utlenianie w wysokich temperaturach, a także lepszą stabilnością strukturalną, co pozwala na ich długotrwałe użytkowanie bez istotnej utraty właściwości mechanicznych. Dodatkowo, stale te wykazują dobrą spawalność, co umożliwia ich szerokie zastosowanie w konstrukcjach przemysłowych, takich jak kotły, rurociągi, zbiorniki ciśnieniowe oraz inne elementy pracujące pod obciążeniem cieplnym i mechanicznym.

Właściwości tych stali są ściśle powiązane z procesami obróbki cieplnej, takimi jak normalizacja, hartowanie czy odpuszczanie, które zapewniają odpowiednią mikrostrukturę i zdolność materiału do pracy w ekstremalnych warunkach. Ponadto, stale te muszą spełniać rygorystyczne normy dotyczące wytrzymałości, wydłużenia i odporności na pełzanie.

W celu walidacji metody SPT zespół, w którym pracował doktorant, wraz z Urzędem Dozoru Technicznego (UDT) podjął kroki w pozyskaniu stali najczęściej stosowanych w przemyśle. Planowo, dzięki walidacji nowej, eksperymentalnej metody badawczej, UDT zyska możliwość korelacji badań ukierunkowanych na ocenę stanu konstrukcji w najbardziej wymagających zastosowaniach i poprawy ogólnych warunków bezpieczeństwa eksploatacji.

W laboratorium przeanalizowano łącznie 9 rodzajów materiałów w różnych stadiach eksploatacji. Zbiór materiałów przedstawiono w tabeli 3.1. Materiały były opisane kodem dla łatwiejszej referencji w trakcie eksperymentów badawczych. Fragmenty materiałów do badań pochodziły z aktywnie wykorzystywanych konstrukcji, takich jak kotły parowe oraz rurociągi. Materiały wykorzystywane były do walidacji metody dla danego materiału oraz na poczet badań pod kątem niniejszej pracy, gdzie skupiono się przede wszystkim na trzech materiałach, tj. 10H2M, 13HMF oraz 15HM.

Materiały tej kategorii charakteryzują się zbliżonym składem chemicznym (Tablica 3.2) oraz uzyskują podobne wartości głównych parametrów materiałowych tj. granicy plastyczności i granicy wytrzymałości. Zestawienie tych parametrów dla konkretnych stali przedstawiono w tabeli 3.3. Wartości parametrów dla materiałów w stanie nieulepszonym cieplnie.

Stal 15HM (13CrMo4-5) [144] to gatunek niskostopowy, przeznaczony do pracy w wysokich temperaturach, wyróżniający się niską zawartością węgla w porównaniu do innych stopów o podobnym zastosowaniu. Charakteryzuje się dobrą plastycznością zarówno w warunkach obróbki na zimno, jak i na gorąco, a także wysoką obrabialnością i spawalnością. Zachowuje wysoką wytrzymałość mechaniczną zarówno w temperaturze pokojowej, jak i podwyższonej. Dzięki odporności na działanie agresywnych środowisk, w tym pary wodnej zawierającej różne związki chemiczne, jest wykorzystywana w budowie zbiorników ciśnieniowych, turbin, kotłów oraz rurociągów pracujących do temperatury 530°C. Elementy wykonane z tej stali obejmują dennice, kołnierze, kolana, trójniki, rury bezszwowe, pręty i odkuwki, a także części rurociągów i przegrzewaczy pary. Materiał dostarczany jest w stanie normalizowanym i odpuszczonym, a także w stanie ulepszonym cieplnie oraz wyżarzonym zmiękczająco.

Stal 10H2M (10CrMo9-10) [144] to perlityczna stal stopowa, cechująca się wysoką odpornością na pełzanie i przeznaczona do pracy w warunkach podwyższonej temperatury, nieprzekraczającej 580°C. Pro-

Material	Kod	Eksplatacja [tys. godz.]
10H2M	D1	0
	J3	30
	J1	249
	K2-K-W	300
	K7-K-W	300
13HMF	M1	0
	M2	0
	M3	0
	R1	200
	R2	217
	R3	280
	K2-K-S	300
	K7-K-S	300
15HM	H1	0
	T1	130
15NCuMNb	N5	0
	N1	0
18CuNMT	C2	140
	C1	207
K22M	W1	350
21HMF	J2	X
SA516 Gr70	B2	0
	B3	0
St37.2/S235JR rurociąg PERN	B1	X

Tabela 3.1: Zbiór materiałów badanych w laboratorium SPT w Centrum Energetyki AGH wraz z oznaczeniem kodowym oraz czasem eksploatacji danej próbki.

Material	C	Si	Mn	P	S	Cr	Mo	Ni	Cu	V	Nb	Al	N	Ti
10H2M	0.08–0.15	0.15–0.50	0.40–0.60	≤0.03	≤0.03	2.00–2.50	0.90–1.10	≤0.30	≤0.25	—	—	—	—	—
13HMF	0.10–0.18	0.15–0.35	0.40–0.70	≤0.04	≤0.04	0.3–0.6	0.5–0.65	≤0.3	≤0.25	0.22–0.35	—	≤0.02	—	—
15HM	0.11–0.18	≤0.35	0.40–0.70	≤0.025	≤0.01	0.70–1.15	0.40–0.55	≤0.35	≤0.25	—	—	—	≤0.012	—
15NCuMNb	≤0.17	0.25–0.50	0.80–1.20	≤0.035	≤0.035	≤0.3	0.25–0.40	1.0–1.3	0.5–0.8	—	0.02–0.06	≤0.015	—	—
18CuNMT	≤0.2	0.30–0.50	0.7–1.0	≤0.04	≤0.04	≤0.3	0.25–0.35	1.0–1.2	0.9–1.2	—	—	0.03–0.08	—	0.03–0.05
K22M (20M)	0.17–0.24	0.17–0.37	0.35–0.65	≤0.035	≤0.04	≤0.25	—	≤0.30	≤0.30	—	—	—	—	—
21HMF	0.17–0.25	0.3–0.6	0.3–0.5	≤0.035	≤0.035	1.2–1.5	1.0–1.2	≤0.6	—	0.25–0.35	—	—	—	—
SA516 Gr70	≤0.30	0.13–0.45	0.85–1.20	≤0.035	≤0.035	—	—	—	—	—	—	—	—	—
St37.2/S235JR	≤0.17	—	≤1.40	≤0.035	≤0.035	—	—	—	≤0.55	—	—	—	≤0.12	—

Tabela 3.2: Procentowy skład chemiczny badanych materiałów. [144, 138, 92, 158]

Materiał	Granica plastyczności [MPa]	Granica wytrzymałości [MPa]
10H2M	>300	480-630
13HMF	>300	460-610
15HM	>300	450-600
15NCuMNb	>420	590-780
18CuNMT	>400	550-680
K22M (20M)	>245	>410
21HMF	>550	700-850
SA516 Gr70	>260	>485
St37.2/S235JR	>235	360-510

Tabela 3.3: Właściwości materiałów badanych w laboratorium SPT w stanie nowym normalizowanym. [144, 138, 92, 158]

dukowana jest w formie blach, odkuwek, prętów kutech, rur gorącowałcowanych oraz elementów armatury przemysłowej. Jej główne zastosowania obejmują produkcję części kotłów parowych, rur przegrzewaczych, przewodowych i kotłowych, a także elementów turbin oraz zbiorników ciśnieniowych. W przemyśle chemicznym i energetycznym wykorzystywana jest do budowy węzownic, komór przegrzewaczy, schładzaczy pary i innych krytycznych komponentów. Charakterystyczną cechą tej stali jest obecność węgla M23C6 oraz zwiększona odporność na działanie wodoru w środowiskach średniego ciśnienia do 500°C. Standardowo występuje w stanie ulepszonym cieplnie lub normalizowanym.

Stal 13HMF (14MoV6-3) [144] jest wysokowytrzymałą stalą chromowo-molibdenową, przeznaczoną do pracy w temperaturach 480–550°C, gdzie występuje intensywne oddziaływanie pary i gazów przemysłowych. Ze względu na swoje właściwości jest szeroko stosowana w rurociągach, częściach kotłów oraz turbin. W stanie normalizowanym wytwarzane są z niej rury bezszwowe do zastosowań w instalacjach ciśnieniowych i parowych. Istotnym dodatkiem stopowym jest wanad, który poprawia właściwości wytrzymałościowe, a także zwiększa odporność na pękanie w porównaniu do innych stali kotłowych. Typowe produkty wykonywane z tego materiału to łopatki turbin, tarcze kierownicze, elementy zbiorników ciśnieniowych oraz elementy armatury rurociągowej, takie jak kołnierze, trójniki czy łączniki. Dostarczana jest w postaci prętów kutech, odkuwek i rur bezszwowych.

Stal 15NCuMNb (15NiCuMoNb5-6-4) [144] to niskostopowy materiał konstrukcyjny o podwyższonej zawartości niklu i miedzi, przeznaczony do pracy w temperaturach do 500°C. Wykazuje wysoką odporność na utlenianie i działanie gorącej pary, co czyni ją szczególnie przydatną w konstrukcjach kotłów, rurociągach oraz aparaturze reaktorów jądrowych. Zastosowanie tej stali obejmuje budowę walczków kotłów pracujących do 375°C, zbiorników ciśnieniowych, elementów wysokoprężnych instalacji energetycznych oraz urządzeń przemysłu chemicznego. W porównaniu do innych materiałów wykazuje bardzo dobrą spawalność oraz odporność na zmienne warunki eksploatacyjne. Standardowo jest dostarczana w postaci blach gorącowałcowanych, rur bezszwowych, tulei oraz prętów kutech i walcowanych.

Stal 21HMF (21CrMoV5-7) [144], należy do grupy stali kotłowych przeznaczonych do pracy w wysokich temperaturach, szczególnie w przemyśle energetycznym. Stosowana jest w konstrukcji elementów

turbin parowych, części wirników, tarcz wirnikowych oraz pokryw zbiorników pracujących w temperaturach do 550°C. Dzięki odpowiednio dobranym dodatkom stopowym, w tym chromowi, molibdenowi i wanadowi, stal ta cechuje się wysoką wytrzymałością mechaniczną oraz stabilnością mikrostruktury w długotrwałej eksploatacji. Znajduje zastosowanie w budowie wałów turbinowych, elementów złącznych, śrub, nakrętek i kołnierzy. Dostarczana jest głównie w formie odkuwek, prętów oraz elementów kutych, dostosowanych do pracy w warunkach intensywnego oddziaływania temperatury i ciśnienia.

Stal SA516 Gr70 [138] to stal węglowa o drobnoziarnistej strukturze, przeznaczona do pracy w podwyższonych temperaturach. Charakteryzuje się dobrą spawalnością i plastycznością, co czyni ją odpowiednią do zastosowań w przemyśle energetycznym i chemicznym. Wykorzystywana jest głównie do produkcji zbiorników ciśnieniowych, kotłów oraz innych urządzeń pracujących pod ciśnieniem. Dostarczana jest w postaci blach o różnych grubościach, które są następnie formowane i spawane w celu uzyskania pożądanych kształtów i rozmiarów.

Stal St37.2 [92], znana również jako S235JR, to stal konstrukcyjna o niskiej zawartości węgla, charakteryzująca się dobrą spawalnością i plastycznością. Jest powszechnie stosowana w budownictwie oraz inżynierii mechanicznej do konstrukcji nośnych, elementów mostów, budynków, a także w przemyśle maszynowym. Dostępna w różnych formach, takich jak blachy, profile, rury i pręty, co umożliwia jej wszechstronne zastosowanie w różnych gałęziach przemysłu.

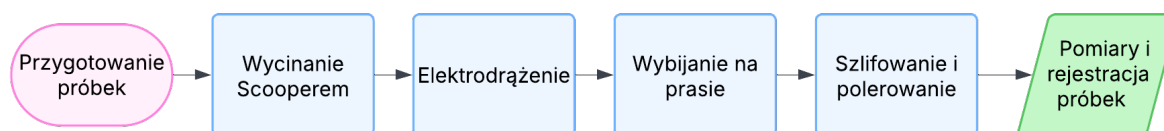
Stal 18CuNMT [158] to stal niskostopowa z dodatkiem miedzi, niklu i molibdenu, charakteryzująca się podwyższoną wytrzymałością i dobrą odpornością na korozję atmosferyczną. Dzięki swoim właściwościom jest stosowana w konstrukcjach narażonych na działanie czynników atmosferycznych, takich jak mosty, wieże czy konstrukcje morskie. Dostarczana jest w postaci blach, prętów i profili, które są wykorzystywane w różnych aplikacjach inżynierskich.

Stal K22M [92], znana również jako 20M, to stal kotłowa o zawartości chromu i molibdenu, przeznaczona do pracy w podwyższonych temperaturach. Charakteryzuje się dobrą odpornością na pełzanie oraz korozję w środowiskach zawierających parę wodną i gazy spalinowe. Stosowana jest w produkcji elementów kotłów, rurociągów parowych oraz innych urządzeń energetycznych pracujących w wysokich temperaturach. Dostarczana w formie blach, rur oraz odkuwek, które są następnie przetwarzane na komponenty instalacji przemysłowych.

3.2 Przygotowanie próbek

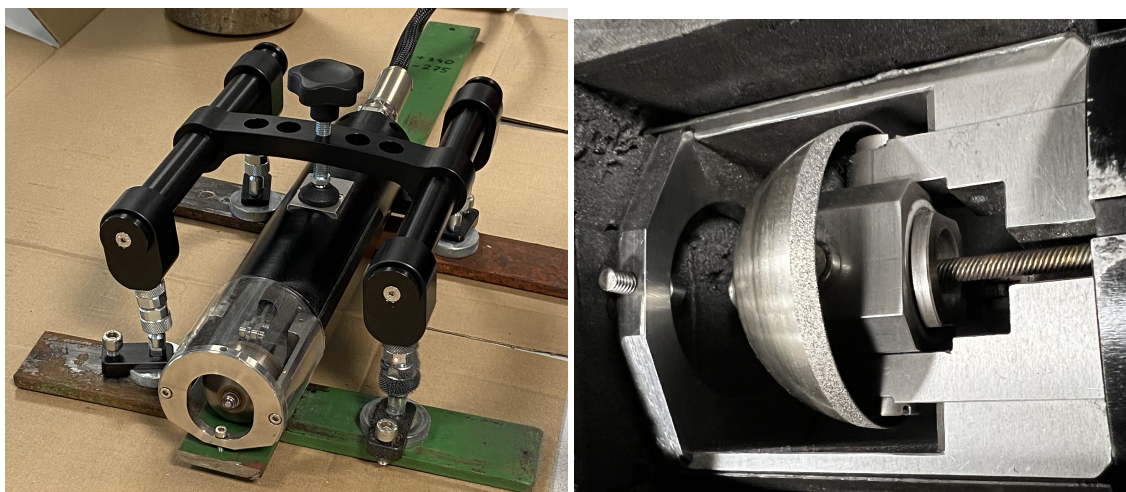
Podczas prac eksperymentalnych wykonywanych w laboratorium SPT znajdującym się w Centrum Energetyki AGH, do analiz oraz przygotowania próbek z materiałów badawczych wykorzystywano szereg urządzeń oraz metod obróbki materiału, aby przejść cały ciąg od badanej konstrukcji do otrzymania wyników SPT. Proces przygotowania został przedstawiony na schemacie blokowym (Rysunek 3.2) oraz opisany szczegółowo w dalszej części podrozdziału.

Materiały opisane w poprzedniej sekcji pozyskane do badań miały różne formy, począwszy od fragmentów rurociągów zawierających cały pusty cylinder w postaci krótkiej rury, po bloki ścian zbiornika ciśnieniowego. Są to jednak przykłady będące ułatwieniem problemu, który w warunkach przemysłowych jest bardziej skomplikowany, ponieważ badane konstrukcje często są na stałe przytwierdzone, zabudowane,



Rysunek 3.2: Schemat blokowy przygotowania próbek SPT

wspawane lub niemożliwe do ingerowania w strukturę w konwencjonalny sposób. W ramach przygotowania do aplikowania metod SPT w przemyśle na wyposażenie laboratorium w Centrum Energetyki AGH pozyskano urządzenie zgodne z normą [18] oraz metodologią pobierania próbek przywołaną na rysunku 2.2 w rozdziale 2. Urządzenie zwane Scooper pozwala na pobieranie próbek bezpośrednio z powierzchni badanej konstrukcji poprzez przytwierdzenie urządzenia bezpośrednio do obiektu będącego źródłem próbki (Rysunek 3.3). Scooper przytwierdzić można do rurociągu bądź kotła z pomocą magnetycznych kotw, a w przypadku rur o mniejszej średnicy z pomocą pasów zaciskowych.

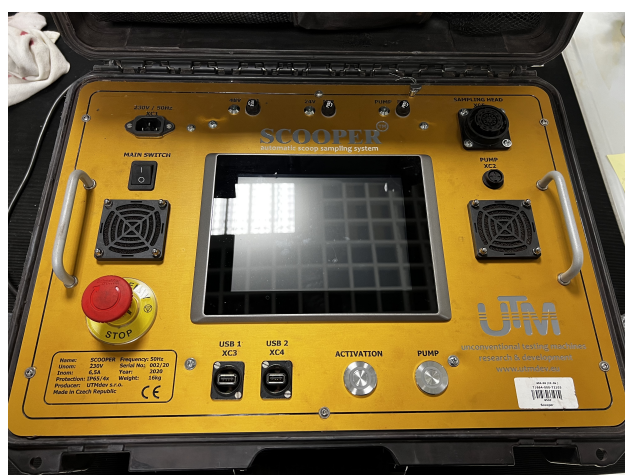


Rysunek 3.3: Proces pobierania próbki

Proces pobierania próbki metodą Scoop Cutter Sampling (SCS) podzielić można na następujące etapy:

1. Określenie lokalizacji pobierania próbek na danym obiekcie lub jego fragmencie oraz zapewnienie dostępu do źródła zasilania elektrycznego (230V/5A) dla urządzenia Scooper.
2. Zamocowanie urządzenia na obiekcie badań przy użyciu pasów mocujących lub magnetycznego uchwytu dociskającego, w zależności od kształtu i gabarytów badanego elementu.
3. Podłączenie urządzenia do sieci elektrycznej.
4. Uruchomienie jednostki sterującej (Rysunek 3.4) oraz oprogramowania sterującego procedurę wycinania próbek (Scooper Control). System monitoruje parametry procesu w czasie rzeczywistym i umożliwia ich dostrojenie.
5. Praca urządzenia w trybie automatycznym lub manualnym.

- W trybie automatycznym system dynamicznie dostosowuje parametry procesu w zależności od oporu stawianego przez obrabiany materiał, co minimalizuje ryzyko uszkodzenia urządzenia.
 - W trybie manualnym konieczne jest podanie parametrów, takich jak: prędkość obrotowa wału [rpm], prędkość zmiany kąta natarcia [$^{\circ}$ /min] oraz wartość kąta natarcia, przy której proces zostanie zakończony.
6. Zakończenie pracy urządzenia następuje samoczynnie po wykryciu braku oporu obrabianego materiału lub manualnie poprzez zatrzymanie programu.
 7. Pobranie próbki z uchwytu magnetycznego lub jej ręczne oddzielenie od badanego materiału w zależności od wybranej metody.
 8. Umieszczenie pobranej próbki w woreczku strunowym, oznaczonym informacjami dotyczącymi miejsca pobrania, numeru próbki, kodu lub nazwy materiału oraz daty pobrania.
 9. Tak zabezpieczone próbki mogą być zostać wykorzystane do przygotowania próbek testowych.



Rysunek 3.4: Jednostka sterująca urządzenia tnącego.

Narzędziem tnącym podczas procesu wycinania jest rotująca stalowa czasza o średnicy 50 mm i grubości 0.3 mm na krawędzi tnącej, pokryta krystalicznym azotkiem boru (z ang. Cubic Boron Nitride - CBN) (Rysunek 3.5). Czasze są elementem zużywającym się podczas pobierania próbek i w zależności od twardości badanego materiału mogą wystarczać na 7-8 pobrań materiału, zanim muszą zostać wymienione ze względu na zużycie pokrycia ciernego na krawędzi tnącej czaszy. Posuw oraz obroty wirnika czaszy kontrolowane są przez jednostkę centralną. Czasza w warunkach pracy obraca się z prędkością 10 000 obrotów na minutę. Materiał jest sukcesywnie podbierany, aż próbka w kształcie ścinka sfery lub łezki w przekroju oddzieli się od materiału bazowego, a następnie mocowana jest przez magnes, w przypadku próbek ze stali o właściwościach magnetycznych, lub jest nacinana do momentu pozostania niewielkiej ilości materiału łączącego, a następnie odłamywana.

Próbki wycięte przez Scooper mają około 30 mm średnicy i 3-5 mm grubości, zależnie od kąta natarcia czaszy na materiał. Próbki w tym stanie nie nadają się jeszcze do obróbki w laboratorium i muszą zostać



Rysunek 3.5: Czasza tnąca



Rysunek 3.6: Próbką materiału w kolejnych etapach przygotowania: a) lezka pobrana z materiału bazowego; b) płaski plaster wycięty z lezki; c) pozostałość materiału po wybiciu próbek SPT na prasie.

przygotowane u zewnętrznego dostawcy usług poprzez cięcie ich w procesie elektrodrażenia na dyski o grubości około 0.8 mm. Dopiero tak pocięte próbki nadają się do kolejnego etapu przygotowania. Aby uzyskać średnice próbki 8 mm, określoną w normie [18], w laboratorium SPT wykorzystuje się manualną pracę hydrauliczną, gdzie w matrycy, z pomocą stempla, wybija się próbki z wcześniej przygotowanych dysków materiału. Średni uzysk próbek do testów to 5-7 sztuk zdalnych do dalszej obróbki. Uzysk próbek zależy od kąta natarcia czaszy, wpływającego na końcową grubość lezki oraz jakości próbki po cięciu jej na plastry. Na rysunku 3.6 przedstawiono przykładową próbkę.

Próbki wybite na prasie ręcznej mają domyślną średnicę 8 mm, a ich grubość, w zakresie około 0.8 mm, dalece odbiega od wymaganego 0.5 mm. W tym celu próbki szlifuje się na szlifierce stołowej będącej na wyposażeniu laboratorium. Szlifierka operuje w zakresie od 50 do 600 RPM ze zmiennym kierunkiem obrotów. Wyposażona jest w dwa dyski szlifujące o średnicy 12 cali oraz niezależne krany z bieżącą wodą dla każdego z dysków, zapewniające nawilżenie powierzchni cierniej podczas szlifowania.

Szlifowanie odbywa się z wykorzystaniem specjalnych narzędzi przystosowanych do próbek SPT składających się z zestawu pierścieni dystansowych oraz obciążnika z ceramiczną podstawą trzymającego próbkę w trakcie pracy. Kolejne pierścienie mają zmienną grubość i pozwalają na szlifowanie w etapach, co zwiększa dokładność wykonanych próbek. W skład pierścieni wchodzi pierścień kalibracyjny oraz dodat-

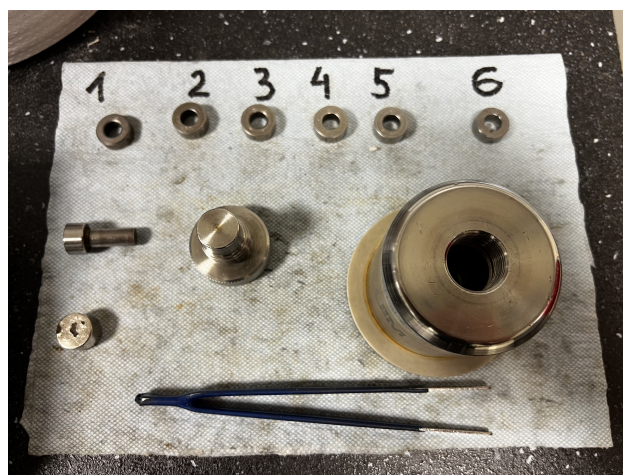


Rysunek 3.7: Szlifierka laboratoryjna

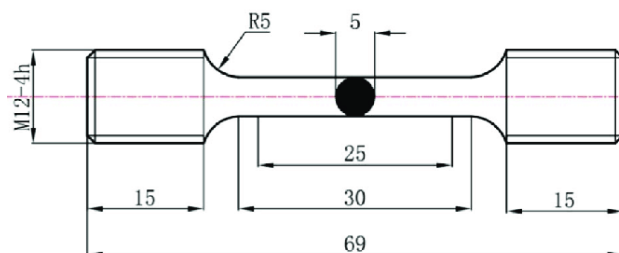
kowy pierścieni pomocniczy, ułatwiający osiągnięcie wymaganej grubości w przypadku materiałów trudnościeralnych. W trakcie szlifowania wykorzystuje się papier ścierny od gradacji 240, poprzez pośredni 320, aż do finalnego polerowania na papierze o gradacji 1200.

Próbki szlifowane są do momentu uzyskania grubości 0.5 mm z tolerancją do ± 0.005 mm. W celu weryfikacji grubości pomiar wykonywany jest na kalibrowanym mikrometrze. Kluczowym dla dokładności pomiaru jest, aby dokładnie osuszyć i oczyścić próbkę z pomocą niepylącego czyściwa, aby zniwelować wpływ obcych cząstek na powierzchni próbki w momencie pomiaru. Tak przygotowane próbki są gotowe do wykorzystania w testach SPT oraz SPCT.

Osobnym procesem jest przygotowanie próbek do badania konwencjonalnego tj. próby jednoosiowego rozciągania (UTT). W tym celu z materiałów opisywanych w poprzedniej sekcji również przygotowywano



Rysunek 3.8: Narzędzia do szlifowania próbek SPT: zestaw pierścieni, kamień szlifierski, dociski.



Rysunek 3.9: Schemat próbki do badania UTT

próbki standardowe. Proces ten odbywał się w pełni w ośrodkach zewnętrznych, ze względu na dużą ilość materiału potrzebną do przeprowadzenia badań. Z trzech głównych materiałów, na których skupiano się w dalszej części badań, wycięto materiał na próbki do badań w ilości satysfakcjonującej do statystycznej oceny wyników tj. minimum 30. Materiały te to 10H2M, 13HMF oraz 15HM. Próbkę wykonano według normy ISO [50], przyjmując kształt obszaru badawczego jako wałek, aby otrzymać kołowy przekrój poprzeczny (Rysunek 3.9). Średnica wałka w analizowanym obszarze wynosiła $d_0 = 5$ mm z dokładnością do ± 0.1 mm, a przyjęta długość obszaru pomiarowego wynosiła $L_0 = 20$ mm z dokładnością do ± 0.2 mm.

3.3 Procedura eksperymentalna

3.3.1 SPT

Badanie SPT w Laboratorium SPT w Centrum Energetyki AGH odbywa się na maszynie wytrzymałościowej TTS 190 (Rysunek 3.10) zgodnie z normą [18]. Test polega na stopniowym wywieraniu nacisku na próbkę SPT poprzez stempel o półkulistym zakończeniu. Stempel działa siłą prostopadle do płaszczyzny próbki. Zabieg wgniatania próbki można porównać do testów twardości, jednak w przypadku SPT kluczowym parametrem jest ugięcie próbki u , ze względu na fakt stopniowego pocieniania samej próbki poprzez rozciąganie materiału będące wynikiem stałego posuwu stempla. Stan naprężenia w teście, w przeciwieństwie do metody klasycznego jednoosiowego rozciągania, jest wieloosiowy. Zakończenie testu jest automatyczne po wykryciu spadku siły poniżej 0.8 wartości F_m , gdzie F_m jest maksymalną siłą w teście, co jest związane z pęknięciem próbki.

Maszyna testowa składa się z:

- uniwersalnej maszyny testowej z kontrolowanym posuwem z głowicą o zakresie pracy do 10 kN;
- komory kriogenicznej do testów w niskich temperaturach;
- zbiornika Dewara na ciekły azot z próbnikiem do chłodzenia komory kriogenicznej;
- trzpienia testowego do mocowania próbek oraz jednostki sterującej.

Elementy maszyny TTS 190 kluczowe dla właściwego przeprowadzenia badania to:

- Indukcyjny sensor typu Keyance GT12 z dokładnością pomiaru do ± 0.001 mm na którym odbywa się pomiar ugięcia próbki.

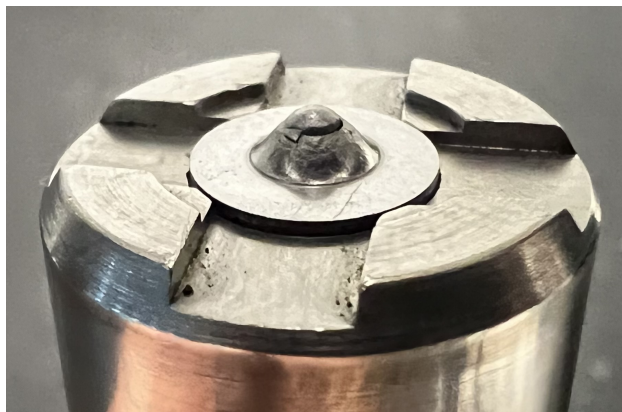


Rysunek 3.10: Maszyna wytrzymałościowa TTS 190

- Stempel matrycy górnej, zakończony półkulą, o promieniu $r = 1.25mm$. Stempel naciska na próbkę generując jej ugięcie w kierunku matrycy dolnej.
- Matryca dolna, pierścieniowa wspiera próbkę podczas testu. Matryca dolna charakteryzuje się otworem o średnicy $D = 4mm$, wspierając próbkę SPT po zewnętrznym obwodzie, pozostawiając miejsce na ugięcie próbki w otworze.

Badanie SPT w laboratorium AGH odbywa się w następujących krokach:

1. Pomiar grubości próbki badanej i zanotowanie wyniku.
2. Uruchomienie maszyny testowej oraz komputera sterującego wraz z oprogramowaniem.
3. Uruchomienie programu testowego z wymaganym posuwem 0.5 mm/min .
4. Wprowadzenie informacji wejściowych na temat próbki:
 - Grubość próbki [mm]
 - Temperatura w czasie testu [C]
 - Materiał - tylko do klasyfikacji
5. Umieszczenie próbki w matrycy testowej.
6. Umieszczenie matrycy testowej w trzpieniu testowym i zakręcenie jej na gwincie.
7. Uruchomienie procedury testowej.
8. Automatyczne ustawienie głowicy w pozycji gotowej do testu.



Rysunek 3.11: Próbką na matrycy, wyjęta z maszyny testowej SPT

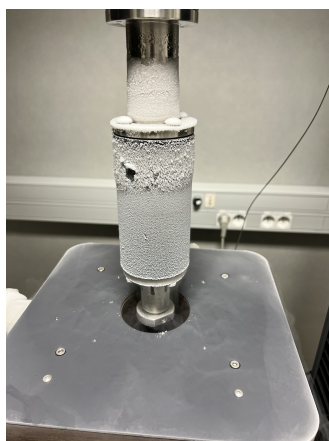
9. Potwierdzenie rozpoczęcia badania wraz z pomiarem siły oraz ugięcie próbki.
10. Zakończenie testu.
11. Zapisanie wyników w pliku tekstowym.
12. Wyjęcie próbki SPT z matrycy testowej (Rysunek 3.11).

Z danych wejściowych grubość próbki oraz temperatura służą jedynie do właściwego zapisu danych końcowych z testu w nazwie pliku zawierającego wyniki. W pliku znajdują się następujące dane numeryczne, zmiennoprzecinkowe wyjściowe w formacie UTF-8, tekstowym kolumnowym, gdzie separatorem dziesiętnym jest przecinek:

- Siła [N]
- Ugięcie powierzchni próbki [mm]
- Czas testu [t]
- Przesunięcie głowicy [mm]

W przypadku wykonywania badań w temperaturze niższej niż pokojowa wykorzystuje się moduł zarządzający temperaturą w komorze kriogenicznej. Komora dostosowana jest do utrzymywania temperatury do -190°C , co umożliwia przeprowadzanie badań pod kątem temperatury przejścia w stan kruchy. W czasie takiego badania, pomiędzy krokiem 6 a 7 w procedurze przeprowadzania badania SPT, dodaje się kolejne kroki:

1. Podłączenie zbiornika Dewara z ciekłym azotem.
2. Uruchomienie jednostki kontrolnej odpowiedzialnej za sterowanie temperaturą.
3. Zadanie temperatury testu w oprogramowaniu.
4. Po osiągnięciu wymaganej temperatury, przejście do standardowych etapów testu (Rysunek 3.12).



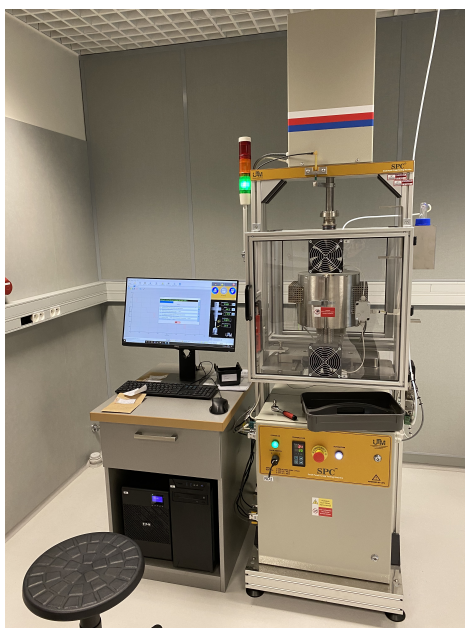
Rysunek 3.12: Komora kriogeniczna w czasie badania SPT w niskich temperaturach.

3.3.2 SPCT

SPCT jest wariantem badania SPT skupiającym się na właściwościach materiału pod kątem odporności na pełzanie w warunkach stale oddziaływającego obciążenia oraz podwyższonej temperatury. Oba te parametry decydują o wyniku końcowym badania. Ze względu na metodę SPCT będącą wciąż w fazie rozwoju, korelacja z wynikami badania klasycznego jest wciąż badana. Ze względu na brak unifikacji badań klasycznych, wprowadzany jest dodatkowy stopień komplikacji pod kątem oceny wpływu parametrów wejściowych badań standardowych na badanie SPCT. W przypadku temperatury, możliwość przekształcenia nie jest trudna, jednak dla obciążenia wciąż przyjmuje się uproszczone założenia. W badaniu klasycznym obciążenie próbki w trakcie testu jest określone jako obciążenie rozciągające wywierane na próbkę w jednej osi. Dla badania SPCT mówimy o naprężeniu wieloosiowym zadawanym tylko przez stempel, lub w przypadku badań w Centrum Energetyki AGH, stalowy stempel naciskający na ceramiczną kulkę o średnicy 2.4 mm wykonaną z azotku krzemu. Aby uniknąć skomplikowanych obliczeń naprężeń, do testu podaje się jedynie siłę nacisku stempla. W literaturze [43, 18] przedstawiono bazową wartość współczynnika Chakrabarty'ego $\Psi = 1,890 \text{ N/MPa}$, która pozwala przybliżyć translację danego obciążenia $[N]$ zadanego podczas badania SPCT dla standardowej geometrii próbki SPT na naprężenie do testu klasycznego $[MPa]$ na tym samym materiale w tej samej temperaturze. Istnieje wariant pozwalający na dalszą optymalizację tej wartości, jednak ze względu na potrzebę korelacji konkretnych wyników w sposób empiryczny przyjęto podstawowy stosunek. Do badań eksperymentalnych, ze względu na przedstawione wcześniej wyniki, otrzymane dzięki uprzejmości Instytutu Metalurgii Żelaza im. Stanisława Staszica w Gliwicach, stosowano proste przekształcenie tego równania (3.1), wyznaczając obciążenie konieczne do zadania próbce $[N]$ na podstawie znanej wartości w $[MPa]$ dla klasycznego testu, a następnie porównywano czas potrzebny do zerwania próbki w teście klasycznym z czasem potrzebnym do przebicia próbki przez ceramiczną kulkę w teście SPCT.

$$F = 1.89 \times \sigma \quad (3.1)$$

Test w laboratorium SPT w Centrum Energetyki AGH odbywa się na maszynie SPC 800 DLS 3.13 składającej się z komputera z dodatkowym źródłem zasilania awaryjnego, pieca do utrzymywania temperatury w trakcie testu oraz systemu obciążenia.



Rysunek 3.13: Maszyna SPC 800

Kluczowe parametry techniczne maszyny SPC 800:

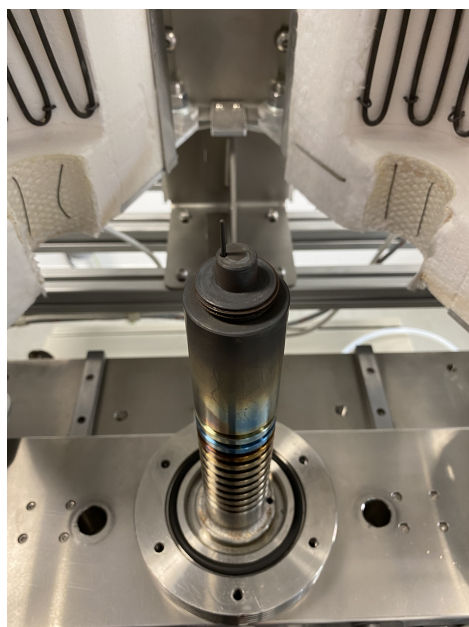
- Zakres temperatur pracy: Temperatura pokojowa - 800C
- Indukcyjny czujnik przemieszczenia o rozdzielczości 0.001 mm
- Stały pomiar temperatury próbki poprzez termoparę umieszczoną przy gnieździe w matrycy testowej
- Atmosfera ochronna argonu
- Czujnik przepływu masy gazu ochronnego

Procedura przeprowadzenia testu:

1. Uruchomienie jednostki sterującej maszyny oraz komputera.
2. Umieszczenie próbki SPCT w kartridżu (Rysunek 3.14)
3. Nałożenie kartridżu na trzpień zamocowany w maszynie testowej w obszarze pieca grzewczego (Rysunek 3.15)
4. Przykręcenie trzpienia do podstawy maszyny.
5. Zamknięcie osłony bezpieczeństwa.
6. Odkręcenie zaworu źródła argonu (butla lub zawór systemowy) i ustawienie ciśnienia wejścia 50 kPa.
7. Wprowadzenie wymaganych parametrów testu oraz próbki
 - Temperatura testu [C];
 - Obciążenie próbki w trakcie testu [N];



Rysunek 3.14: Kartridż na próbkę do badania SPCT



Rysunek 3.15: Trzpień testowy maszyny TTS 190

- Grubość próbki [mm];
- Opis referencyjny próbki: materiał, numer testu, dodatkowe notatki.

8. Rozpoczęcie procedury testowej.

9. Zakończenie testu i wyjęcie próbki po schłodzeniu.

Wynikiem badania po jego zakończeniu jest czas do pęknięcia próbki, który jest głównym punktem porównawczym z klasycznym badaniem oraz współczynnik pełzania.

3.3.3 Klasyczne badanie UTT

Próba jednoosiowego rozciągania do celów badania wyników metod SPT i UTT oraz porównawczych z metodą SPT była wykonywana na Wydziale Inżynierii Metali i Informatyki Przemysłowej AGH. Próby wykonywane były na maszynie wytrzymałościowej Zwick/Roell Z250. Maszyna wyposażona jest w uchwyty do rozciągania próbek okrągłych o uchwytych gwintowanych zgodnych ze standardem rynkowym. Parametry pracy maszyny podczas testu:

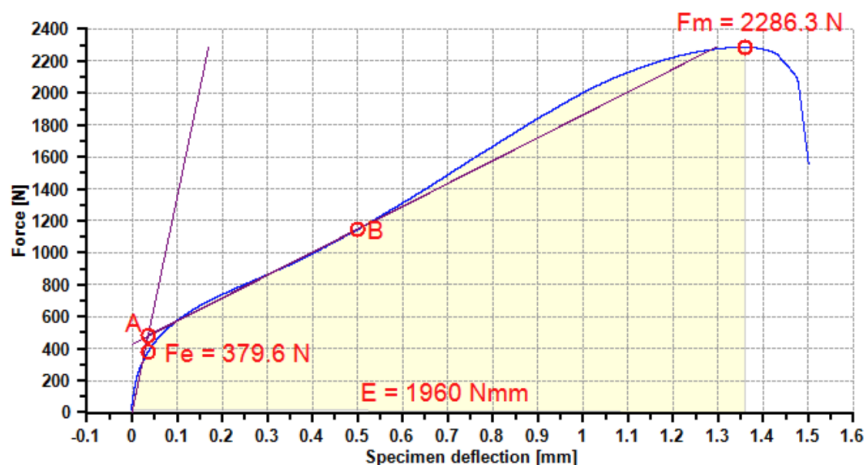
- Maksymalna siła rozciągająca: 250 kN
- Siła wstępna w teście: 800 N
- Prędkość modułu E: 0.0025 1/s
- Prędkość badania: 0.005 1/s

3.3.4 Analiza mikroskopowa

Po zakończeniu analizy statystycznej, aby dalej potwierdzić założenia przyjęte na początku badania materiału 10H2M, mówiące o jego nieprzerwanej użyteczności do zastosowania przemysłowego w stanie zmęczonym, przeprowadzono analizę mikrostrukturalną materiału klasy J1 przy użyciu skaningowego mikroskopu elektronowego LEO 1450-05-19 (3705/TR) na próbkach przekątnych, które zostały mechanicznie szlifowane i polerowane, a następnie trawione odczynnikami Mi1Fe zgodnie z normą PN-61/H-04503. Analiza została przeprowadzona dla warstw wewnętrznej, zewnętrznej oraz środkowej pierwotnego materiału.

Skaningowa mikroskopia elektronowa (SEM) jest doskonałym narzędziem do analizy topografii powierzchni przełomów. Mikropęknięcia można identyfikować poprzez obserwację powierzchni przełomu, na której mogą występować charakterystyczne cechy, takie jak fasety rozdzielcze, dolki plastyczne czy struktury „rzeki”. Fasety rozdzielcze to zazwyczaj płaskie i gładkie obszary, natomiast dolki plastyczne to małe, okrągłe wgłębienia, świadczące o plastycznym charakterze przełomu. Struktury „rzeki” to rozgałęzione linie, które można prześledzić wstecz do miejsca inicjacji pęknięcia.

Przeprowadzono ocenę degradacji materiału oraz poszukiwanie wymienionych wyżej wskaźników wizualnych zgodnie z „Wytycznymi Urzędu Dozoru Technicznego nr 1/2015. Zasady diagnostyki i oceny trwałości eksploatacyjnej elementów kotłów i rurociągów pracujących w warunkach pełzania (2015)” [140]. Ocenę przeprowadzili pracownicy Urzędu Dozoru Technicznego (UDT) w Polsce.



Rysunek 3.16: Przykładowa krzywa SPT

3.4 Metody analizy wyników eksperymentalnych

3.4.1 Interpretacja danych

Badanie SPT w swoim założeniu ma na celu zastąpienie badania klasycznego, a tym samym z założenia powinno prezentować takie same wyniki, z uwzględnieniem niepewności statystycznej. Zbudowanie krzywej naprężenie-odkształcenie jest dla SPT w standardowym podejściu niemożliwe, jednak niniejsza praca poszerza ten horyzont, pozwalając dla konkretnych materiałów odtworzyć standardową krzywą oraz z niej pozyskać parametry materiałowe badanych stali. Dla wspomnianego standardowego podejścia do badania SPT, na podstawie parametrów wygenerowanych w trakcie testu, tj. Siła $F[N]$ oraz ugięcie próbki $u[mm]$ wyrysowana jest krzywa tożsama z wykresem zależności siły od przemieszczenia w badaniach klasycznych, jednak o właściwym dla SPT przebiegu (Rysunek 3.16). Widoczny jest tam krótki, jednak wyraźny odcinek sprężysty, przechodzący w długi zakres plastyczny.

Na podstawie tak otrzymanych krzywych, a przede wszystkim na podstawie wyników surowych, nieobrobionych, dla parametrów oprzyrządowania testowego w laboratorium SPT w Centrum Energetyki AGH, wykonuje się obliczenia opisywane szerzej w rozdziale 2, w sekcji 2.9, mające na celu wyznaczenie granicy wytrzymałości R_m (wzór 2.21, gdzie $\beta_{R_m} = 0.192$) oraz granicy plastyczności R_{P02} (wzór 2.18 oraz 2.23, gdzie $\beta_{R_{P02}} = 0.510$) a także wartości energii testu E_{SP} (wzór 2.20) równej powierzchni pola pod wykresem siła-ugięcie i służącej do określania odporności na pękanie próbki, wykorzystywanej przede wszystkim w badaniu temperatury przejścia w stan kruchy. Przejście ze stanu sprężystego w plastyczny jest charakteryzowane poprzez wartości siły F_e wyznaczanej metodą bi-linearną poprzez dopasowanie dwóch przecinających się krzywych na wykresie, o odczytanej dla wartości u_e . Wszystkie wzory te mają swoje źródło w normie EN 10371:2021 [18].

Wyniki otrzymywane, zgodnie z powyższą metodologią, uzyskano dla każdej z próbek materiału. Podobnie w przypadku metody klasycznej UTT. Wyniki, jakie zapisywano do porównania, to dwa główne, wspomniane wcześniej parametry: granica wytrzymałości R_m oraz granica plastyczności R_{P02} . Parametry wyznaczone obydwojema metodami badawczymi dla materiałów testowych przyjętych w tej pracy przeana-

lizowano statystycznie według metod przedstawionych w podrozdziale 3.4.2 niniejszej pracy. Opisano tam statystyczne zestawienie rozkładów wyników SPT oraz UTT, a dalej w pracy też wyników z sieci neuro nowej (NN) celem weryfikacji jakości wyników wszystkich metod. Granica plastyczności i granica wytrzymałości to podstawowe i kluczowe parametry wykorzystywane w doborze materiału do konstrukcji i te parametry są także kontrolowane w czasie eksploatacji konstrukcji i urządzeń.

3.4.2 Statystyczna analiza wyników badań porównawczych

Jako narzędzia statystyczne do porównania oraz jednoczesnego zestawienia metod stosowano przede wszystkim trzy główne metody statystycznej analizy danych. Zastosowano komplementarne wobec siebie metody statystyczne: hierarchiczną analizę skupień z konstrukcją dendrogramów, wykresy prawdopodobieństwa ukierunkowane głównie na ocenę rozkładu, a także histogramy danych stosowane głównie w raportach walidacyjnych metod na potrzeby Urzędu Dozoru Technicznego. Przy wykorzystaniu metod analizy zwrócono uwagę na występowanie wartości odstających. Widoczne są one na graficznych przedstawieniach danych, takich jak ej, gdzie interpretacja ich występowania jest arbitralna. Istnieją jednak metody wyznaczania i określania ich występowania z wykorzystaniem narzędzi statystycznych. Tak też w niniejszej pracy w kolejnych etapach przyjęto metody ich wykluczania. W przypadku analizy klastrow grupowanie danych dla danego stopnia podobieństwa pozwala wyróżnić grupy danych, które nie pasują do pozostałych, poprzez umieszczenie ich w osobnym klastrze niż pozostałe wartości zbliżone, np. do wartości średnich danej grupy. Dla metod takich jak histogramy oraz wykresy podobieństwa analizujące statystyczny i parametryczny rozkład, przed wykonaniem analizy wstępnie przefiltrowano dane surowe z wykorzystaniem klasyfikatorów punktów odstających z zastosowaniem metody Z-score, jednak w zmodyfikowanej wersji, bardziej adekwatnej do typu analizowanych danych. Klasyczny Z-score opiera się na wartości średniej oraz na odchyleniu standardowym. Gdy rozkład jest skośny lub zawiera mocno odstające punkty, będące naturalnie występującymi wartościami związanymi m.in. z niejednorodnością materiału badanego, wartości odchylenia standardowego zmodyfikowany Z-score spełni funkcję klasyfikatora danych odstających dużo lepiej, zwłaszcza dla wielkości badanej podgrupy w zakresie 30-60 indywidualnych wartości danych. Przy wyznaczaniu zmodyfikowanego Z-score wykorzystuje się parametr *MAD* (*Median Absolute Deviation*) [45], czyli miarę rozproszenia zdefiniowaną jako mediana bezwzględnych odchyłeń obserwacji od mediany populacji, wyznaczana ze wzoru (3.2). Dalej parametr *MAD* podstawiany jest do wzoru na właściwy zmodyfikowany Z-score (3.3) [151] z którego otrzymana wartość porównywana jest z przyjętym progiem klasyfikacji danej wartości jako punkt odstający. W inżynierii materiałowej dla małej populacji próbek wartością bazową jest 3.5. Jeśli M_i jest większe od wartości progu klasyfikacji, to daną obserwację uznaje się za punkt odstający i jest ona dyskwalifikowana z dalszej analizy statystycznej, aby nie zaburzać wyników. Takie podejście pozwala połączyć odporność analizy na błędy wynikające z metody lub czynnika ludzkiego z prostotą i akceptacją naturalnego rozkładu danych materiałowych.

$$MAD = \text{median}(|x_i - \tilde{x}|), \quad i = 1, \dots, n \quad (3.2)$$

gdzie:

- x_i – i -ta obserwacja w próbce,

- $\tilde{x} = \text{median}(x_1, \dots, x_n)$ – mediana całej populacji,
- n – liczebność populacji.

$$M_i = 0.6745 \frac{x_i - \tilde{x}}{\text{MAD}} \quad (3.3)$$

gdzie:

- M_i – zmodyfikowany wynik z dla obserwacji x_i ,
- 0.6745 – stały współczynnik skalujący (dla danych normalnych)
- MAD - mediana bezwzględnych odchyleń obserwacji od mediany populacji.

W celu określenia najlepiej dopasowanego rozkładu dla każdej grupy pomiarowej, co wykorzystywane było dalej w konstruowaniu przedstawianych wykresów, wykonano obliczenia dla następujących parametrów, przeprowadzając test jakości danych [93]:

- Statystyka dopasowania Andersona-Darlinga (AD) - Statystyka AD stanowi ważoną miarę odchylenia skumulowanego rozkładu empirycznego $F_n(x)$ od rozkładu teoretycznego $F(x)$. W stosunku do testu Kolmogorova-Smirnova nadaje ona większe wagi obserwacjom w ogonach rozkładu, co czyni ją szczególnie czułą na różnice w obszarach newralgicznych z punktu widzenia inżynierskich zastosowań (np. ocena wytrzymałości materiałów w testach Small Punch). Kryterium minimalnej wartości AD pozwala ustawić ranking rozkładów według malejącego dopasowania: im niższa wartość, tym mniejsze globalne odstępstwo danych od modelu.
- Wartość p (P) dla testu Andersona-Darlinga (AD). Wartość p interpretuje się jako prawdopodobieństwo uzyskania statystyki AD co najmniej tak ekstremalnej jak obserwowana, przy założeniu, że badany rozkład idealnie opisuje dane (hipoteza zerowa H_0). Zachowanie wysokiej wartości p (zwykle $p > 0,05$ lub surowsze progi przy licznych testach równoległych) wskazuje, że brak jest statystycznych przesłanek do odrzucenia H_0 , co wspiera dalsze wykorzystywanie danego modelu rozkładu w analizach inferencyjnych. Jednocześnie wartość p uzupełnia informację o samej wielkości statystyki AD o aspekt losowej zmienności, dzięki czemu pomaga uniknąć przewagi rozkładów „nadmiernie dopasowanych” jedynie wskutek fluktuacji próby. W praktyce zastosowane podejście „dual-metric” – minimalizacja AD przy równoczesnym wymaganiu, aby odpowiadająca jej wartość p była nie mniejsza niż ustalony poziom istotności – pozwala na zbalansowanie kryteriów dobroci dopasowania i odporności na nadmierne dopasowanie. Taka procedura jest rekomendowana w literaturze metrologicznej i materiałowej, gdzie próby bywają niewielkie, a rozkłady często charakteryzują się ciężkimi lub skośnymi ogonami [89].

Histogramy stanowią graficzną ilustrację częstości występowania poszczególnych wartości lub zakresów wartości w badanej próbie, co pozwala na szybkie zidentyfikowanie kluczowych cech rozkładu danych [151, 120]:

- **Kształt rozkładu:** Rozkład może być symetryczny, skośny w prawo, dłuższy „ogon” po prawej stronie, lub w lewo, kiedy ogon jest po lewej stronie lub rozkład Weibulla.
- **Wielomodalność:** Obecność kilku wyraźnych maksimów (szczytów) sugeruje, że populacja danych może być niejednorodna i składa się z co najmniej dwóch podgrup o odmiennych parametrach.
- **Wartości odstające:** Skrajnie niska lub wysoka liczba obserwacji w odległych „koszykach” (przedziałach) histogramu może wskazywać na istnienie wyników o odstających wartościach, które mogą wymagać pogłębionej analizy pod kątem przyczyn (np. błąd pomiaru, nietypowe warunki eksperymentu).

W analizach naukowych histogram często stanowi punkt wyjścia do bardziej zaawansowanych metod statystycznych. Pozwala też na wstępną weryfikację hipotezy o rozkładzie normalnym. Klasyczny kształt dzwonowy, jeśli nie jest normalny, to wskazuje to na potrzebę rozważenia innych modeli teoretycznych (np. rozkład lognormalny, wykładniczy).

Hierarchiczna analiza skupień (ang. Hierarchical Cluster Analysis), która stanowi metodę służącą do grupowania obserwacji lub zmiennych w zbiory, tzw. klastry, w sposób odzwierciedlający ich wzajemne podobieństwo. Jest to wszechstronna technika znajdująca zastosowanie w różnych obszarach, takich jak biologia, nauki społeczne, marketing czy finanse, wspierając takie zadania jak analiza ekspresji genów, segmentacja klientów czy wykrywanie anomalii [64, 60, 132, 116]. W konsekwencji analiza skupień stanowi nieodzowne narzędzie wspomagające podejmowanie decyzji opartych na danych oraz generowanie hipotez, co przyczynia się do głębszego zrozumienia złożonych zjawisk i rozwoju wiedzy naukowej. Jest to również idealne podejście do porównania metod badawczych. Następnie proces ten jest wizualizowany w postaci dendrogramu. Do budowy dendrogramów i analizy zastosowane podejście wykorzystujące metodę największej odległości. W niej odległość pomiędzy dwoma klastrami definiuje się przez wartość maksymalną spośród wszystkich odległości między poszczególnymi elementami obu klastrów wyrażaną jako $distcomplete(A, B)$. Jeżeli klastry A i B zawierają odpowiednio elementy a_i oraz b_j , to wzór (3.4) opisuje tę zależność [33]:

$$distcomplete(A, B) = \max_{a_i \in A, b_j \in B} (dist(a_i, b_j)) \quad (3.4)$$

Gdzie: A i B to dwa zbiory danych w formie klastrów, złożone z pojedynczych punktów; a_i oraz b_i to pojedyncze obserwacje w klastrach; $dist(...)$ to funkcja odległości pomiędzy obserwacjami; max to największa odległość z pośród wszystkich par punktów pomiędzy klastrami;

Dalej zastosowano metodę wyznaczania odległości Euklidesowej [151], która stanowi jedno z najbardziej intuicyjnych ujęć miary różnic pomiędzy punktami w przestrzeni wielowymiarowej. Jej wzór można wyrazić następująco 3.5:

$$d_{Eucl}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (3.5)$$

gdzie $x = (x_1, x_2, \dots, x_n)$ $y = (y_1, y_2, \dots, y_n)$ są wektorami reprezentującymi wartości analizowanych zmiennych. Za wyborem tej metryki często przemawia jej powszechność i łatwość interpretacji geometrycznej.

Odległość euklidesowa jest powszechnie uznawana za najbardziej intuicyjną miarę dystansu w przestrzeni wielowymiarowej, co przekłada się na łatwość interpretacji i analizowania wyników badań eksperymentalnych, jednak R. Wilcox podkreśla, że może być bardzo czuła na wartości odstające od zbiorów [151]. Ujmuje ona różnice między obserwacjami w sposób zgodny z tradycyjnym podejściem geometrycznym, a przy tym jest spójna z licznymi metodami statystycznymi opartymi na sumie kwadratów błędów. Dzięki temu możliwe jest bezpośrednie i przejrzyste odwzorowanie zmian wektorów cech, co pozwala na lepsze zrozumienie wzajemnych relacji między analizowanymi punktami danych.

Struktura dendrogramu składa się z liści, gałęzi i węzłów. Każdy liść symbolizuje pojedynczy punkt danych, natomiast gałęzie łączą liście i węzły, przedstawiając proces łączenia klastrów. Węzły oznaczają miejsca, w których gałęzie się łączą, co sygnalizuje tworzenie klastrów wyższego rzędu poprzez scalanie klastrów niższego rzędu. Wysokość gałęzi w dendrogramie odzwierciedla stopień niepodobieństwa między łączonymi klastrami – im większa wysokość, tym większa różnica między klastrami. Wysokości te są obliczane na podstawie wybranej miary odległości oraz zastosowanej metody łączenia [93]. W praktyce przekłada się to na tworzenie bardziej zwartych skupień, gdyż już pojedynczy przypadek dużego dystansu pomiędzy elementami dwóch klastrów może zadecydować o braku ich scalenia na danym etapie aglomeracji. Metoda ta jest szczególnie użyteczna, gdy priorytetem jest wyodrębnienie grup o wyraźnie małym zróżnicowaniu wewnętrznym.

Metoda aglomeracji największej odległości zapewnia z kolei większą zwartość wyodrębnianych klastrów, wymagając, by również najbardziej odległe obserwacje w grupie były relatywnie blisko siebie. Taka rygorystyczna definicja dystansu pomiędzy skupieniami ogranicza ryzyko tworzenia sztucznych łańcuchów, charakterystycznych w metodzie *single linkage*, i przeciwdziała łączeniu się danych o niejednorodnych właściwościach w jeden klaster. W konsekwencji dendrogramy opracowane na bazie metody największej odległości wykazują wyraźne różnice między populacjami, co sprzyja klarownej segmentacji obserwacji w kontekście celów badawczych.

Połączenie tej metryki z metodą największej odległości stanowi wyważony kompromis między skutecznością detekcji naturalnych podgrup w danych a zachowaniem jasnej, geometrycznej interpretacji wyników. Umożliwia wyodrębnianie homogenicznych skupień, lepiej pasujących do hipotez formułowanych w badaniach empirycznych, zwłaszcza tych, w których istotne jest jednoznaczne oddzielenie próbek znacząco różniących się.

Wyniki hierarchicznej analizy skupień przedstawiono w postaci dendrogramu, w którym oś pozioma (lub pionowa) odzwierciedla miarę odległości (lub podobieństwa), zaś węzły drzewa wskazują kolejne etapy aglomeracji klastrów. Przyjęty pułap 50% podobieństwa pełni rolę arbitralnej granicy, powyżej której uznaje się, że analizowane obserwacje tworzą relatywnie jednorodny klaster.

W kontekście badań naukowych takie grupowanie pozwala ocenić, czy dane próbki bądź wyniki z różnych metod badawczych są do siebie zbliżone w stopniu umożliwiającym potraktowanie ich jako jednej

populacji, czy raczej wykazują na tyle duże różnice, że należy je wyraźnie rozróżniać w sensie statystycznym.

Wykresy podobieństw (z ang. Probability Plot) służące do wizualnej oceny, czy rozkład empiryczny danych jest zbliżony z określonym rozkładem teoretycznym [120]. Uporządkowane wartości obserwacji lub statystyki pozycyjne są porównywane z teoretycznymi kwantylami wynikającymi z wybranego modelu. Jeśli punkty obserwowane układają się w przybliżeniu wzdłuż linii prostej, uznaje się, że analizowany rozkład jest przydatny do opisu otrzymywanych danych.

Rozkład normalny (Gaussa) to jedno z najczęściej przyjmowanych założeń w analizach statystycznych, opisany funkcją gęstości prawdopodobieństwa (3.6) [120]:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (3.6)$$

gdzie μ to średnia, a σ – odchylenie standardowe populacji.

W kontekście wykresów podobieństwa oszacowane parametry pochodzą z danych empirycznych, natomiast odchylenia punktów od linii odniesienia pozwalają wnioskować o ewentualnych nienormalnościach rozkładu. ramach formalnego testu zgodności z rozkładem normalnym często wykorzystuje się statystykę Andersona–Darlinga (AD). Przy niskiej wartości AD (i jednocześnie wysokim poziomie p-value) wnioskować można, że nie ma podstaw do odrzucenia hipotezy o normalności danych. Tak też w jednej z analiz zaobserwowano, że dane mogą podlegać rozkładowi 2-parametrowemu wykładniczemu, zwanym też ang. *shifted exponential*. Jest on definiowany przez dwa parametry: skalę θ oraz przesunięcie γ . Funkcja gęstości rozkładu (PDF) wyraża się następująco, dla $x \geq \gamma$ (wzór 3.7) [120]:

$$f(x) = \frac{1}{\theta} \exp\left(-\frac{x-\gamma}{\theta}\right) \quad (3.7)$$

Jeżeli $\gamma = 0$, otrzymujemy standardowy rozkład wykładniczy jednoparametrowy. W praktyce statystycznej rozkład ten bywa stosowany wtedy, gdy dane są dodatnie i istnieje pewna wartość progowa, na przykład minimalna, poniżej której obserwacje nie występują. Następnie, analogicznie jak przy rozkładzie normalnym, wartości empiryczne nanoszone są na układ współrzędnych, gdzie oś teoretyczna odwołuje się do kwantyli rozkładu 2-parametrowego wykładniczego. Jeżeli punkty układają się w linii prostej, potwierdza to dopasowanie do tego typu modelu.

3.5 Analiza komputerowa i automatyzacja oceny stanu materiału

W niniejszej sekcji przedstawiono wszystkie działania w zakresie badań materiałowych w kontekście niniejszej pracy, gdzie wykorzystywano kod komputerowy przygotowany jako narzędzie pomocnicze lub docelowe mające za zadanie wypełnienie celów pracy oraz odpowiedź na pytania, czy możliwa jest automatyzacja oceny stanu materiału z wykorzystaniem metody SPT. Kod komputerowy napisany został w języku Python 3.12, a przedstawione skrypty gotowe są do wykorzystania bezpośrednio bez wcześniejszego przygotowania i stanowią kompletny kod pozwalający na odtworzenie ścieżki od surowych danych do osiągnięcia wyników opisanych w niniejszej pracy. Cały kod źródłowy znajduje się w aneksie A do niniejszej

pracy. Przedstawienie to ma na celu podkreślenie kompletności kroków podjętych w ramach badań nad hipotezą pracy.

W przedstawionych skryptach wykorzystywano następujące biblioteki naukowe oraz analityczne:

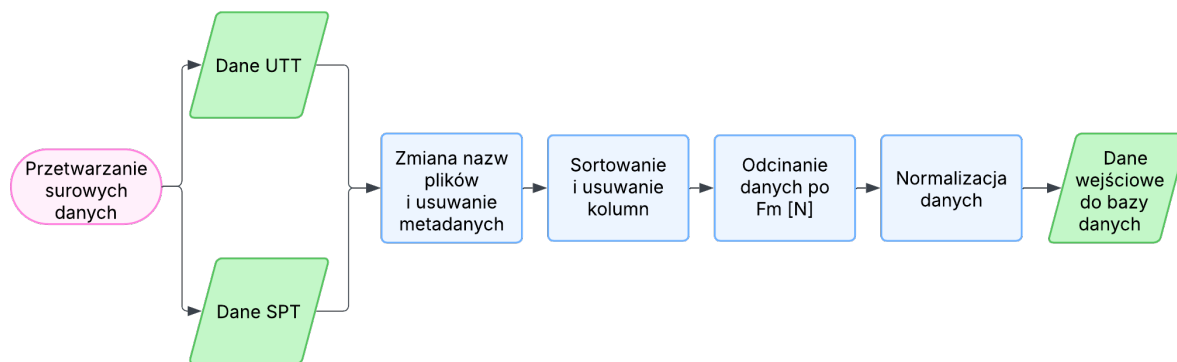
1. Podstawowe biblioteki języka Python [106, 108, 107, 110, 111, 109, 112]:

- *os* - używana we wszystkich skryptach w celu przemieszczania się w systemie plików, do tworzenia folderów lub komponowania ścieżek odczytu danych. Jest to podstawowa biblioteka do właściwej obsługi danych wejściowych i wyjściowych.
- *json* - biblioteka wymagana do zapisywania i przetwarzania danych takich jak parametry sieci neuronowej lub serializacja danych X,Y dla krzywej materiałowej wyjściowej i wyjściowej do bazy danych SQLite. Pozwala na przejście od zapisu punktowego danych do wielowymiarowych wektorów.
- *random* - używana do mieszania wartości identyfikujących próbki podczas przygotowywania paczek danych do nauczania sieci neuronowej. Wykorzystuj kod numeryczny *nasiona* (z ang. *seed*) co identyfikuje losowy zbiór i podział. Zapewnia sprawiedliwy oraz losowy dla użytkownika podział na zestaw do nauczania, walidacja oraz testowania wyników modelu sieci neuronowej co wypełnia naukowe założenia.
- *hashlib* - pozwala na generowanie unikatowych numerów identyfikacyjnych dla danych podczas nauczania sieci neuronowej zapewniając, że podczas podziału na zestawy do uczenia unikatowa próbka występująca w bazie danych wielokrotnie w różnych konfiguracjach nigdy nie znajdzie się, z żadną ze swoich konfiguracji w więcej niż jednym zestawie do którego została przyporządkowana. Ponownie wypełnia to założenie podziału na dane testowe, które nigdy nie brały udziału w nauczaniu i stanowiąc będą weryfikacją sieci.
- *time* - używana pomocniczo do oceny czasu trwania uczenia a by efektywnie planować prace nad siecią i mieć kontrolę nad przewidywanym czasem zakończenia działań. Używana przy przedstawianiu stanu danej operacji w terminalu zintegrowanego środowiska programistycznego.
- *datetime* - podobnie jak *time* używana pomocniczo w raportowaniu i śledzeniu postępu prac

2. Biblioteki wykorzystywane bezpośrednio do prac nad siecią neuronową [114, 4].

- *torch* - centralna platforma programistyczna używana do operacji na tensorach, budowania modelu sieci neuronowych
- *torch.nn* - zapewnia komponenty do budowy architektury modeli sieci neuronowej jak warstwy, dodatkowe funkcje takie jak *dropout* i wykorzystywany jest w każdym skrypcie gdzie pracowania z modelami sieci neuronowych.
- *torch.nn.functional* - dostarcza funkcje aktywacji do sieci neuronowej pozwalające na modyfikacje działania sieci i jej zachowania podczas uczenia ze szczególnym uwzględnieniem zachowań w przypadku błędnych (negatywnych) interpretacji wartości przez neurony.

- *torch.utils.data* - moduł używany do przygotowywania paczek danych oraz ich podział i procesowanie przed i podczas nauczania sieci neuronowej. Pozwala na pobieranie danych wejściowych i wyjściowych z bazy danych i zgodnie z instrukcjami zawartymi w kodzie procesowanie tych danych oraz podawanie ich w odpowiednim momencie do modelu sieci neuronowej w czasie nauczania, walidacja i testowania.
 - *torch.optim* - zawiera algorytmy optymalizacyjne do modeli sieci neuronowych pozwalające na właściwe jej nauczanie. Przykłady to *Adam* i *AdamW* wykorzystywane w niniejszej pracy.
 - *torch.optim.lr_scheduler* - moduł pozwala na odpowiednie planowanie zmian wartości tempa nauczania modelu sieci neuronowej aby zoptymalizować je w momentach kiedy model nie czyni postępów przy obecnych warunkach nauczania.
 - *optuna* - to platforma programistyczna pozwalająca na implementacje nauczania maszynowego w zakresie doboru optymalnych parametrów w zależności od obranego celu. Wykorzystanie tej biblioteki pozwala na implementację gotowego rozwiązania. Dzięki wykorzystaniu tej biblioteki w zakresie pracy zoptymalizowano dobór parametrów sieci neuronowej takich jak rozmiar warstwy sieci, parametr funkcji *dropout*, rozmiar paczki danych do uczenia, prędkość uczenia, aukcję aktywacji oraz dodatkowe parametry wpływające na odporność sieci na zakłócenia pochodzące z danych źródłowych.
3. Biblioteki wykorzystywane do analizy danych i obliczeń numerycznych [98, 102, 124, 125, 46, 137, 4].
- *numpy* - biblioteka pozwalająca na prace nad zbiorami danych liczbowych, używana do wszystkich numerycznych operacji takich jak gradienty, interpolacje, próbkowanie, normalizacja i przygotowanie sygnału, obliczenia statystyczne, wektoryzacja operacji, generowanie przestrzeni wartości dla funkcji gęstości prawdopodobieństwa czy przekształcenia danych
 - *pandas* - to biblioteka służąca do manipulacji i analizy danych tabelarycznych. W skryptach tego typu wykorzystywana jest do wczytywania, filtrowania, agregowania i zapisu danych eksperymentalnych. Stanowi główny interfejs między danymi wejściowymi (np. z plików CSV lub Excel) a analizą statystyczną
 - *sqlite3* - biblioteka wymagana do odczytu i zapisu oraz pracy ze scentralizowaną bazą danych w formacie SQLite.
 - *openpyxl* - służąca do odczytu i zapisu plików w formacie .xlsx (Excel). W projektach naukowych umożliwia automatyczne tworzenie i aktualizację plików wynikowych zawierających dane statystyczne, bez potrzeby ręcznej edycji arkuszy kalkulacyjnych.
 - *matplotlib* - to podstawowa biblioteka do wizualizacji danych w języku Python. Moduł umożliwia generowanie wykresów histogramów, PDF (probability density functions), wykresów QQ (quantile-quantile), oraz graficzną reprezentację parametrów statystycznych. Jest nieodzowna w analizie eksperymentalnej, gdyż pozwala na szybkie i przejrzyste porównanie rozkładów danych rzeczywistych i modelowanych.



Rysunek 3.17: Schemat blokowy przygotowania wyników SPT i UTT do implementacji w bazie danych

- *scipy* - dostarcza szeroki zakres narzędzi statystycznych, w tym funkcje dopasowywania rozkładów, testy normalności (np. Shapiro-Wilka, Andersona-Darlinga), obliczenia parametrów rozkładów, funkcje gęstości i dystrybucji. Jest nieodzownym elementem naukowej analizy danych eksperymentalnych.
- *tqdm* - biblioteka usprawniająca pracę z programami, których operacje są czasochłonne. Pozwala na wizualizację postępu prac na przykład w postaci paska postępu pracy.
- *sklearn.model_selection.train_test_split* - używany jest do tworzenia powtarzalnych zestawów danych do nauczania sieci zachowując unikalność próbek pomiędzy zestawami
- *sklearn.linear_model.LinearRegression* - pozwala między innymi na dopasowanie za pomocą liniowej regresji linii odzwierciedlających zakres sprężysty na krzywych Naprężenie-Odształcenie co wykorzystywane jest w analizie danych końcowych i wyznaczaniu własności materiałowych.
- *sklearn.metrics* - moduł dostarcza metryki ewaluacyjnego używane podczas uczenia, walidacji i testowania sieci neuronowych przydane do oceny jakościowej wyników wygenerowanych przez sieć.

3.5.1 Przetwarzanie danych eksperymentalnych

Przygotowanie surowych danych z testów materiałowych do implementacji w bazie danych odbywa się etapami. Proces postępowania został przedstawiony na schemacie blokowym (Rysunek 3.17) oraz opisany dalej w tekście.

Pierwszym krokiem na etapie przygotowania danych do modelowania było oczyszczenie danych źródłowych ze zbędnych zakłóceń oraz błędów, które w normalnym podejściu nie stanowią problemu dla badacza lub operatora maszyny, są jednak niepotrzebnym czynnikiem, który musi zostać usunięty, zanim dane będą załadowane do sieci neuronowej lub bazy danych. W przypadku badania klasycznego dane wejściowe zawierały 5 dodatkowych wierszy metadanych, które nie wносиły wartości dodanej do treści pliku obrabianego pod kątem zbioru uczącego oraz jego zastosowania w sieci neuronowej. Przetworzenie surowych danych eksperymentalnych i usuwanie zakłóceń rozpoczęto od przygotowania zmienionej nazwy plików

tekstowych danych UTT z formy opisowej, nadanej w laboratorium dla ułatwienia komunikacji i transferu danych, na nazwę łatwo rozróżnialną w bazie danych. Następnie, ze względu na niezgodny z dalszą obróbką kod zapisu plików, zmieniono standard Unicode zapisu plików z "windows-1252" na "UTF-8". W przypadku plików z badania UTT wymagane było usunięcie 5 wierszy metadanych oraz zmiana języka nazewnictwa z języka polskiego na angielski. W badaniu SPT należało usunąć dodatkowe kolumny danych, a w obu przypadkach posortować kolumny, aby zunifikować proces implementacji w bazie danych. Istotnym punktem wpływającym na dalsze wyniki prac była decyzja o odcięciu wartości siły surowych danych obu testów w momencie osiągnięcia maksymalnej wartości. Decyzja ta została podjęta na podstawie analizy wizualnej wyników SPT i UTT, aby nie wprowadzać dodatkowych zmiennych mogących pogorszyć efektywność predykcji wyników kluczowych dla celu pracy, czyli granicy plastyczności i granicy wytrzymałości. Wszystkie kroki zostały przedstawione w formie kodu w skrypcie A.1. Przetworzone pliki danych SPT i UTT są gotowe do poddania normalizacji. Normalizacja to kluczowa technika przetwarzania danych, polegająca na skalowaniu cech wejściowych do wspólnego zakresu lub rozkładu. W kontekście uczenia maszynowego i sieci neuronowych normalizacja pełni kilka istotnych funkcji. Po pierwsze, zapewnia, że każda cecha wnosi zbliżony wkład w proces uczenia. Gdy cechy mają bardzo różne skale, na przykład jedna w zakresie od 0 do 10, a inna od 0 do 10 000, cecha o większym zakresie może dominować w aktualizacjach gradientu, co prowadzi do jej nadmiernego uwzględnienia. Poprzez normalizację przekształcamy te cechy do podobnego zakresu, często od 0 do 1 lub od -1 do 1, co sprawia, że proces uczenia się staje się bardziej zrównoważony i efektywny. Dodatkowo, normalizacja znacząco przyspiesza zbieżność algorytmów optymalizacyjnych, takich jak metoda gradientu prostego. Sieci neuronowe opierają się na numerycznych metodach optymalizacyjnych, które są bardzo wrażliwe na skalę danych wejściowych. Dzięki normalizacji gradienty stają się bardziej stabilne, co umożliwia szybszą i bardziej niezawodną zbieżność. Stabilność ta jest szczególnie istotna w głębokim uczeniu, gdzie propagacja gradientów przez wiele warstw może prowadzić do problemów takich jak zanikanie lub eksplozja gradientów, jeśli dane wejściowe nie są odpowiednio skalowane. Normalizacja przyczynia się również do stabilności numerycznej, zapobiegając występowaniu bardzo dużych lub bardzo małych wartości, które mogłyby powodować przepełnienie lub niedomiar w obliczeniach. Jednakże przed przystąpieniem do normalizacji danych ważne jest, aby określić skalę normalizacji, do której będziemy odnosić wszystkie nasze dane. W tym celu przeszukano repozytorium wszystkich danych SPT i UTT wykorzystywanych do tej analizy, aby potwierdzić najwyższe wartości siły oraz wydłużenia bądź ugięcia, w zależności od testu, które dla ułatwienia oceny korelacji opisano w plikach angielskim słowem 'Distance'.

Najwyższe wartości dla plików z testów SPT i UTT:

- Maksymalne wartości dla UTT
 - Force: 13253.1, File: /Users/.../T1 UTT/T1 UTT_F8.TXT
 - Distance: 2.607, File: /Users/.../UTT/J1 UTT/J1 UTT_F4.TXT
- Maksymalne wartości dla SPT
 - Force: 1741.3, File: /Users/.../R1 SPT/501_24C_R1-104.txt
 - Distance: 1.775, File: /Users/.../SPT/H1 SPT/505_21C_3.txt

Na podstawie otrzymanych danych wyjściowych przyjęto wartości maksymalne do normalizacji, a dane poddano procesowi przetwarzania. Skrypt został przygotowany w ten sposób, aby po zmianie jedynie wartości do normalizacji był zdolny do obróbki plików UTT oraz SPT. Dzięki normalizacji wszystkie pliki dla sześciu analizowanych materiałów są gotowe do zapisania w bazie danych. Krok ten odbywał się na kolejnym etapie, którym było sparowanie, agregacja i uporządkowanie surowych, znormalizowanych danych do formy przetwarzanej przez sieć neuronową, co jest wymagane do trenowania. Skrypt A.2 do normalizacji danych przedstawiono w załączniku.

3.5.2 Budowa baz danych z wynikami eksperymentów

Do przygotowania bazy danych próbek przyjęto wstępną filtrację wyników na podstawie zachowań zaobserwowanych na surowych krzywych materiałowych zarówno dla testów SPT, jak i UTT. Liczba próbek ujętych w bazie różni się zatem od liczby próbek poddawanych analizie porównawczej pomiędzy metodami SPT i UTT (Tabela 3.4). Uzasadnienie nieuwzględnienia części próbek to uszkodzone dane, wynikające z problemów podczas przeprowadzania testów. Problemy z próbkami są częstym zjawiskiem, a ich występowanie w całej populacji jest naturalne, co przedstawiono dalej w dziale poświęconym analizie i porównaniu wyników testów. W przypadku bazy danych odsiew wyników potencjalnie mogących niewłaściwie wpływać na proces uczenia, mogą one zaburzać oraz wprowadzać zakłócenia. Załączenie błędnych lub anomalnych wyników, wynikających z błędu maszyny lub niedokładności konkretnego pomiaru, zaburzyłoby zdolność sieci neuronowej do nauczania się rzeczywistego powiązania SPT z UTT, w szczególności przy ograniczonej liczebności zbioru uczącego. Proces uczenia, w szczególności, jest niezwykle wrażliwy na występowanie próbek odstających. Może wpływać na wyniki oceny zdolności sieci do generalizacji wyników i właściwej predykcji. W dalszej implementacji takiego modelu do pełnej sieci automatyzującej proces oceny stanu materiału może nieproporcjonalnie wpłynąć na wynikowy kształt krzywej odkształcenie-napężenie, zawyżając lub zaniżając predykcje albo błędnie wskazując moment zmian, takich jak osiągnięcie stanu plastycznego przez materiał. W praktyce, w przypadku włączenia wyników zawierających błędy w zestawie uczącym, sieć może nauczyć się schematów, które odwzorowują błędy w teście, zamiast konkretnego zachowania danych próbek. Zmniejsza to niezawodność modelu, wpływając na metryki weryfikacyjne oraz zwiększając ryzyko niskiej generalizacji wyników na nieznanach zbiorach danych, takich jak zbiory testowe. Skutkuje to zaniżeniem pewności predykcji danej sieci neuronowej. Dlatego też dołożono wszelkie starania, aby odfiltrowanie danych nienadających się do implementacji w bazie danych wykonano dokładnie, zwracając uwagę na każdą próbkę z osobna. W przypadku badania UTT zmniejszyło to znacząco populację niektórych grup danych. Krok ten miał na celu zachowanie jedynie próbek, które w sposób rzeczywisty odzwierciedlają naturalne wariacje w zachowaniach badanych materiałów z tej samej grupy oraz odchylenia standardowe wynikające z dokładności i czułości danej metody badawczej. Podniosło to jakość zbiorów danych, a w efekcie końcowym zwiększyło pewność predykcji sieci neuronowej. Wszystkie surowe dane oraz raporty z laboratorium dla wyników UTT i SPT, zawierające dane nieodfiltrowane, są dostępne do wglądu oraz oceny w repozytorium [61].

Material	10H2M		13HMF		15HM	
Test	D1	J1	R1	M3	H1	T1
SPT	47	46	32	38	35	33
UTT	31	27	27	28	25	29
Sum of pairs	1457	1242	864	1064	875	957

Tabela 3.4: Zestawienie liczby próbek SPT i UTT zaimplementowanych do bazy danych.

Baza danych *Point to Point*

Pierwszym etapem konstrukcji bazy danych było wykorzystanie zdefiniowanej funkcji *convert_and_clean_data()*, która przetwarzała każdą daną wejściową, zmieniając ciągi numeryczne używające przecinka jako separatora dziesiętnego na liczby zmiennoprzecinkowe w formacie *float32*. Zapewniało to istotne zużycie pamięci operacyjnej w przypadku tak dużej, konstruowanej sieci. Następnym krokiem było sparowanie danych w funkcji *pair_data_based_on_percentiles()*, gdzie w przypadku testów SPT i UTT występowała różnica w liczbie pomiarów, przy czym pliki SPT zawierały znacznie więcej punktów pomiarowych niż pliki UTT. W bazie danych, aby zastosować je do uczenia modelu sieci neuronowej, należało połączyć wyniki prób SPT z odpowiadającymi im wynikami UTT. Ze względów jednak na fakt, że niemożliwym jest przetestowanie dokładnie tej samej próbki obiema metodami, przyjęto zaakceptowaną na początku projektu generalizację zestawiania każdego wyniku SPT dla danego materiału z każdym wynikiem UTT. Ze względu jednak na złożoność zadania, jakim jest parowanie całych przebiegów krzywych materiałowych pochodzących z dwóch różnych testów, w bazie danych wykonano parowanie każdego punktu krzywej SPT z każdym punktem krzywej UTT dla każdej próbki. W efekcie tego zabiegu baza danych, która z sukcesem została wygenerowana dla analizowanej grupy materiałów, miała objętość dyskową 19 GB i przedstawiała 73 miliony wierszy będących parami punktów pomiędzy metodami SPT i UTT. Tak zbudowana baza danych, ze względu na ograniczenia sprzętowe komputerów używanych do obliczeń podczas budowy modelu sieci neuronowej, musiała zostać zmniejszona. To wykazało, że bezpośrednie zestawianie punkt po punkcie było niepraktyczne. Zmniejszenie w każdym przypadku generuje odstępstwo od stanu idealnego i kompletnego, gdzie porzucane są dane potencjalnie ważne, jednak zastosowanie metody opartej na percentylach umożliwiło pobranie próbek z danych SPT w równych odstępach, a następnie sparowanie ich z odpowiadającymi im punktami z danych UTT. Podejście to pozwoliło na zachowanie ogólnego kształtu krzywej oraz reprezentację istotnych cech wykresu, przy jednoczesnym uwzględnieniu różnic w gęstości próbkowania obu zbiorów danych. Przyjęto parowanie co 10 punktów, aby ograniczyć objętość bazy danych, co efektywnie zmniejsza liczbę punktów w danej krzywej do około 1000, co wciąż zapewnia dokładne odzwierciedlenie charakterystyki wyników z obu metod badawczych. Wygenerowana została baza danych o objętości dyskowej 1.3 GB oraz 9 milionach wierszy par. Proces parowania został opisany w funkcji *process_and_pair_folders()*, w której funkcja odczytuje pliki tekstowe ze wskazanych folderów skoordynowanych po nazwach grup materiałów, np. D1, a następnie generuje wszystkie możliwe kombinacje połączeń pomiędzy plikami w obu folderach SPT i UTT. Kolejnym etapem była agregacja sparowanych danych oraz ich zapis w uporządkowanej formie, tj. w

	ID	Material	SPT_File	UTT_File	SPT_Force	SPT_Distance	UTT_Force	UTT_Distance
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
11	11	D1	normalized_500_25C_19.txt	normalized_D1 UTT_F_B9.txt	0,01777777777777778	0,00111111111638860555	0,05688992592592593	1,2902259259259258e-05
12	12	D1	normalized_500_25C_19.txt	normalized_D1 UTT_F_B9.txt	0,020611110263400612	0,0013333333966632221	0,05687562962962963	1,5414777777777776e-05
13	13	D1	normalized_500_25C_19.txt	normalized_D1 UTT_F_B9.txt	0,023555556403266054	0,001611111090622	0,056861999999999996	1,5508592592592592e-05
14	14	D1	normalized_500_25C_19.txt	normalized_D1 UTT_F_B9.txt	0,02638888888888889	0,0018333333233992223	0,05687644444444445	1,5638296296296294e-05
15	15	D1	normalized_500_25C_19.txt	normalized_D1 UTT_F_B9.txt	0,02911111195882161	0,002055555556176389	0,05689940740740741	1,4902888888888889e-05
16	16	D1	normalized_500_25C_19.txt	normalized_D1 UTT_F_B9.txt	0,03188888973659939	0,002277777788953611	0,05690251851851852	1,1238296296296297e-05

Rysunek 3.18: Wycinek struktury bazy danych

bazie danych SQLite. Funkcja ta została przedstawiona jako *insert_data_into_db()*, a skrypt otwierał bazę danych SQLite z dedykowaną tabelą *paired_data*, w której zapisywane były wszystkie metadane danej pary punktów. Agregacja umożliwiła nie tylko systematyczne przechowywanie informacji, ale także ułatwiła dostęp do danych podczas dalszych analiz. W praktyce całość sparowanych danych stanowiła spójny zbiór treningowy, który jest niezbędny do nauki modelu sieci neuronowej – model ten uczył się odwzorowywać relacje pomiędzy danymi SPT a wynikami UTT. Procedura budowy bazy danych została przedstawiona w postaci skryptu umieszczonego w aneksie (Skrypt A.3). Zbiór danych uzyskany w wyniku sparowania i agregacji (Rysunek 3.18) był wykorzystywany w kolejnych etapach.

Baza danych *Curve to Curve*

Alternatywne podejście przyjęto podczas budowy drugiej bazy danych. Metodologia przyjęta w tym wypadku wynikała ze zmiany podejścia w celu poprawy wyników generowanych przez sieć neuronową. Skrypt A.9 generuje ustrukturyzowaną, podobnie jak w przypadku pierwszego wariantu bazy danych SPT-UTT, tablicę zawierającą pary próbek. W tym przypadku jednak zdecydowano się na zastosowanie parowania krzywa do krzywej. W czasie formowania bazy danych całe znormalizowane wykresy SPT są łączone z całymi znormalizowanymi wykresami UTT. W przeciwieństwie do poprzedniej punktowej bazy danych, ta zawiera całe krzywe w postaci zbioru surowych punktów sparowanych w zakresie danej grupy materiału, np. H1, a dane są serializowane jako ciągi danych w formacie *json*. Każdy wiersz w bazie danych odpowiada jednej pełnej parze wyników, opisanych metadanymi w postaci grupy materiału oraz nazw plików źródłowych. Całkowity schemat postępowania odpowiada schematowi tworzenia bazy danych *Point to Point*, opisanej w poprzedniej podsekcji. Tak przygotowana baza danych była gotowa do pracy z alternatywnymi architekturami modeli sieci neuronowych.

Ze względu jednak na ograniczenia wynikające z zasobów potrzebnych do odczytu tak skomplikowanej bazy danych oraz operacji wymaganych do odczytu pełnego ciągu znaków z każdego wiersza bazy danych, a następnie dostosowania go jako danych wejściowych do nauczania, zdecydowano się przygotować mniejszą, roboczą wersję opisanej bazy danych. Skrypt A.10 przekształca uprzednio przygotowaną bazę danych z pełnej długości krzywych do adaptacyjnie zeskalowanych krzywych 300 punktowych, reprezentujących najważniejsze aspekty przebiegu wykresu. Zapewnione jest to funkcją *adaptive_sample_curve()*, w której wykorzystywana jest strategia ważenia zmienności za pomocą wyznaczania pochodnych pierwszego i drugiego rzędu dla sygnału siła-przemieszczenie, co zapewnia dokładniejsze odwzorowanie w obszarach dużej zmienności przebiegu, a mniej precyzyjne odtworzenie tam, gdzie nie jest to konieczne, jak w przypadku szeroko rozciągniętego zakresu plastycznego. Dzięki temu możliwa jest implementacja takiego wektora o

stałej wielkości do sieci neuronowej jako jednej danej wejściowej. Każda oryginalna krzywa, zapisana w formacie *json*, zawiera wciąż wartości znormalizowanej siły oraz znormalizowanego przemieszczenia dla obu testów. Skrypt załadowuje wartości z oryginalnej bazy danych typu *Curve to Curve*, odczytuje każdy wiersz z tablicy *paired_data* i przepuszcza krzywe przez funkcję skalowania. Jako wynik tworzona jest nowa baza danych o takiej samej strukturze, jednak zdecydowanie zmniejszonych rozmiarach i zunifikowanej długości ciągów znaków w każdym z wierszy. Ta baza danych, podobnie jak jej pierwotna wersja, składa się z 6486 rekordów, jednak zajmuje przestrzeń dyskową 166.6 MB w porównaniu do 3.8 GB oryginalnej bazy danych.

3.5.3 Sieć neuronowa

W poszukiwaniu modelu predykcyjnego do budowania korelacji danych z testów UTT i SPT, w pierwszym kroku podjęto decyzję o wyborze środowiska, w którym powstanie projekt. Z dostępnych możliwości przyjęto język programowania Python, w którym do tej pory przygotowywano wszystkie poprzednie kroki. W przypadku środowiska Python niezbędnym narzędziem do budowy nawet prostej sieci neuronowej typu *feedforward* jest pakiet PyTorch. PyTorch jest biblioteką wykorzystywaną w uczeniu maszynowym, pozwalającą na operacje na tensorach, generowanie grafów oraz zawierającą niezbędne funkcje do budowy i pracy z sieciami neuronowymi. Początkowe, wstępne wersje sieci były relatywnie proste, z ograniczoną liczbą warstw ukrytych oraz neuronów, jednak liczne eksperymenty i procesy weryfikacyjne zdecydowały o wykorzystaniu sieci głębokiego uczenia.

Budowa i optymalizacja modelu Sieci Neuronowej z wykorzystaniem uczenia maszynowego

Sieć neuronowa oparta na podejściu *Point to Point* Po serii analiz różnych architektur i wariantów sieci, rozważanych w kontekście celu pracy, gdzie złożoność zagadnienia i natura danych wymagają budowania relatywnie prostych danych wejściowych, składających się z 2 cech pochodzących z testu SPT oraz dwuwymiarowych danych wyjściowych reprezentujących pomiary UTT. Wybrano konfigurację, która dokładnie odwzorowuje nieliniowe zależności pomiędzy danymi, a jednocześnie pozostaje wystarczająco prosta, aby móc ją wykorzystać do przetwarzania danych z pomiarów.

Cechy zastosowanej sieci neuronowej są następujące:

- Model jednokierunkowy (ang. *feedforward*) nazywany jest również MLP (z ang. *multilayer perceptron*) przygotowany do generowania punktów danych krzywych UTT na podstawie wejściowych punktów SPT.
- 4 warstwy, w tym 3 warstwy ukryte, w pełni połączone. Używają one określonej wielkości warstwy, liczonej w neuronach. Ostatnia, czwarta warstwa sprowadza wynik ponownie do 2 wartości: siły i przemieszczenia w domenę UTT. Każda warstwa
- Funkcja aktywacji: Leaky ReLU, a dodatkową funkcją aktywacji jest *dropout*, który jest wywoływany po każdej warstwie ukrytej.

W modelu wykorzystano tzw. świadomość materiału, dodając dodatkowy wektor wejściowy będący wskaźnikiem, z jakim materiałem mamy do czynienia, co odnosi model do danego materiału. To podejście

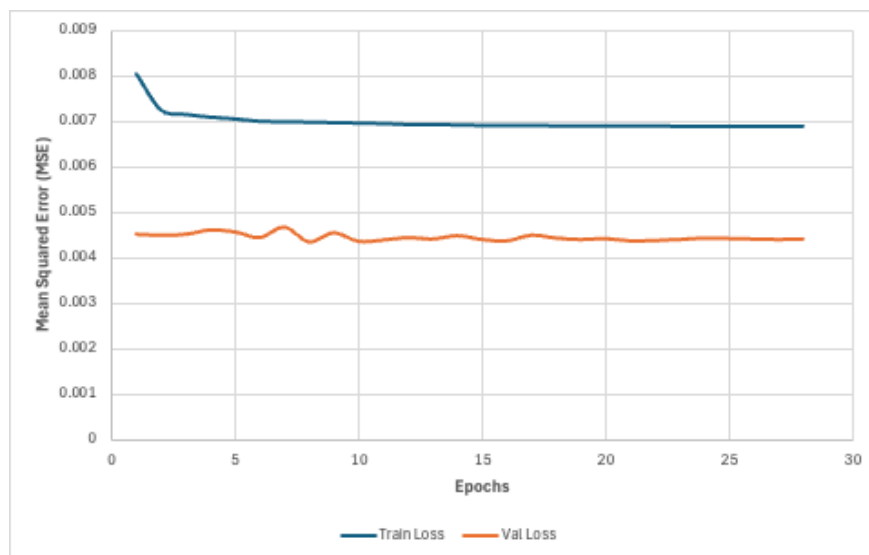
nazywane jest MoE (z ang. *mixture of experts*) sprawdza się przy pracy z odmiennymi od siebie zbiorami danych tego samego typu, jak w przypadku krzywych materiałowych dla różnych stali. Model przyjmuje trzy warstwy ukryte, które pozwalają na hierarchiczną reprezentację, gdzie wczesne warstwy odwzorowują proste zależności i schematy, a warstwy głębsze łączą je w bardziej skomplikowane cechy do generowania na podstawie danych wejściowych. Funkcja *dropout* wykorzystywana jest w celu uniknięcia przeuczenia sieci, co okazało się konieczne ze względu na pozostałe w danych eksperymentalnych oraz naturalnie występujące zakłócenia i szумы, które są nieistotne dla wyniku końcowego. Ostatecznie wykorzystanie funkcji aktywacji Leaky ReLU, w porównaniu do standardowych funkcji, jest uzasadnione koniecznością poprawy jakości modelu poprzez unikanie problemu, w którym neurony kontynuują proces nauczania, nawet jeśli otrzymywane wyniki są negatywne. Ta cecha sieci była jednak poddawana weryfikacji na dalszym etapie pracy, aby wyjaśnić, w jakim stopniu współpracuje ona najlepiej z funkcją *dropout*. Dzięki tak dobranej architekturze uzyskano równowagę pomiędzy dokładnością a prostotą rozwiązania przy ograniczonej liczbie parametrów wejściowych otrzymywanych z testu SPT. Architektura ta stanowi praktyczne rozwiązanie, które, przy dobranych cechach, minimalizuje problem przeuczenia sieci. Architektura ta stanowiła podstawę do wczesnych eksperymentów, a z czasem przerodziła się w jedno z ostatecznie rozpatrywanych rozwiązań.

Aby zoptymalizować model sieci neuronowej w przyjętej formie, określono hiperparametry definiujące go i poddano je procesowi optymalizacji z wykorzystaniem uczenia maszynowego. W tym celu zastosowano bibliotekę Optuna. Optuna to biblioteka w języku Python, zaprojektowana w celu wspierania procesów uczenia maszynowego. Wykorzystuje ona mechanizm poszukiwania najlepiej dobranych kombinacji parametrów z wykorzystaniem Estymatorów Parzena o Strukturze Drzewiastej (TPE, z ang. *Tree-structured Parzen Estimators*)[4]. Proces ten pozwalał systematycznie weryfikować wartości hiperparametrów określonych dla testu, który wykaże minimalny wskaźnik straty podczas walidacji danego zestawu danych. Skrypt do optymalizacji sieci neuronowej A.4 został przygotowany tak, aby wykonać 100 unikatowych prób w zakresie hiperparametrów. Każda próba była wykonywana w 5 epokach, czyli pełnych przejściach przez cały zestaw danych do trenowania, wykorzystując jako metodę oceny wielkość straty (z ang. *validation loss*) mierzoną poprzez błąd średniokwadratowy. W przypadku sieci przygotowywanej w niniejszej pracy była to liczba wystarczająca do określenia najlepszego zbioru hiperparametrów ze względu na dane wejściowe oraz dane wyjściowe, będące prostym wektorem składającym się z wartości siły i przemieszczenia. W przypadku pracy na takich danych oraz architekturze sieci MLP stabilizację osiąga się po 1-2 epokach. Po przejściu wszystkich 100 prób, najlepszy zestaw był zapisywany. Aby zapewnić dane do uczenia, w skrypcie zdefiniowano funkcję *create_data_loaders()* łączącą się z bazą danych *Point to Point* i wyciągającą z niej pary punktów z krzywych SPT i UTT. Dane były dzielone na zestawy danych do uczenia (64%), walidacji (16%) i testowania końcowego (20%). Powtarzalność była zapewniana stałą wartością zmiennej *seed*, używanej do uczenia maszynowego. Każdy podzestaw był ładowany do modelu podczas nauczania w zmiennej, wielkości paczkach, które również podlegały procesowi optymalizacji przez nauczanie maszynowe. Główny człon skryptu do nauczania maszynowego zawarty jest w funkcji *objective()*, która definiuje zakres poszukiwań optymalnych parametrów sieci. Zakresy hiperparametrów zostały dobrane na podstawie ustalonych dobrych praktyk opisanych w literaturze [39, 4]. Informacje o dobieranych parametrach wraz z końcowymi wyni-

kami procesu optymalizacji z wykorzystaniem sieci neuronowej, wraz z zaraportowanymi przez skrypt w oryginalnej formie wartościami, przedstawiono poniżej:

- Prędkość uczenia (ang. *learning rate*) - parametr algorytmu określający, jak duże są zmiany wartości parametrów dokonywane w każdej iteracji. Kontroluje on, jak szybko model aktualizuje wagi parametrów, i gdy dobrana wartość jest zbyt wysoka, model będzie uczył się szybciej, jednak może generować większe rozbieżności:
 - Zakres: $1e-5$ do $1e-1$ - w skali logarytmicznej aby sprawniej weryfikował wpływ wielokrotnie wyższych wartości względem siebie. Sugerowany w literaturze pułap startowy to wartość $1e-03$ [39]
 - Wartość końcowa = $1.6e-05$
- Wielkość warstwy ukrytej - liczba neuronów w każdej warstwie. większe modele mogą reprezentować bardziej złożone funkcje, jednak
 - Zakres: 16, 32, 64, 128, 256 - ze względu na rozmiar bazy danych i przetwarzane wartości wejściowe zdecydowano się dla tego rodzaju sieci parametr ograniczyć w górnym zakresie do 256 neuronów co już klasyfikuje go jako sieć głęboka
 - Wartość końcowa = 256
- Wielkość pakietów danych - liczba przykładów przepuszczanych przez sieć przy każdym przejściu.
 - Zakres: 32, 64, 128, 512 - mniejsze rozmiary pakietów często pomagają we właściwej generalizacji, jednak spowalniają proces uczenia.
 - Wartość końcowa = 64
- Współczynnik *dropout*- wskazujący, ile procent neuronów jest dezaktywowanych przy każdym przejściu.
 - Zakres: 0.1-0.5 - literatura [39] wskazuje na punkt startowy w postaci wartości 0.5 aby unikać przeuczenia jednak optymalna wartość zależna jest od pozostałych współgrających elementów architektury
 - Wartość końcowa = 0.2382
- Wartość krzywej negatywnej - mnożnik negatywnej wartości wyniku neuronu.
 - Zakres: 0.01-0.2 - wyższe wartości pozwalają nadać większą wagę neuronom generującym negatywne wyniki, które w przypadku standardowej funkcji *ReLU* byłyby nieaktywne, co skutkuje poprawą regresji jednak gdyby wartość była zbyt wysoka może zaburzać proces uczenia
 - Wartość końcowa = 0.1591

Na podstawie wyników z prób optymalizacji parametrów przygotowano osobny skrypt (A.5) stworzony do dokładnego uczenia modelu zgodnie z zaproponowanymi parametrami. Skrypt charakteryzuje się limitem 50 epok całkowitego czasu nauczania oraz limitem cierpliwości 20 epok przy braku poprawy wyników walidacji. Korzystano dodatkowo z funkcji *ReduceLROnPlateau* zapewniającej zmniejszanie prędkości uczenia w przypadku braku poprawy po 2 epokach stagnacji. Zapewnia to odpowiednie tempo do stabilizacji procesu uczenia. Te same dane wejściowe zostały podzielone w stosunku 80-10-10, gdzie 80% bazy danych trafiło do zbioru uczącego, a pozostałe 20% danych do zbiorów walidacyjnego oraz ślepego zbioru testowego. Podział zapewniony przez unikalne identyfikatory przypisane do konkretnych krzywych SPT zapewniał brak przenikania danych pomiędzy zbiorami, co jest kluczowe w przypadku bazy danych typu *Point to Point*, gdzie w przeciwnym wypadku model mógłby zobaczyć fragment krzywej przeznaczonej do testowania jeszcze podczas nauki. Model ten, po zakończeniu procesu uczenia, jest następnie zapisywany i gotowy do dalszego wykorzystania w sieci. Po uczeniu model jest dodatkowo testowany na zestawie walidacyjnym i testowym, aby sprawdzić, czy nie został przeuczony. Wyniki strat walidacyjnych przedstawiono na rysunku 3.19. Stanowią one wartości uzyskane po ostatnim etapie uczenia. W tabeli 3.5 przedstawiono



Rysunek 3.19: Krzywe uczenia i walidacji modelu MLP

metryki opisujące zdolności modelu do predykcji, ocenione w czasie testów po procesie uczenia. Są to wartości uśrednione dla wszystkich materiałów, ze względu na charakterystykę bazy danych. Obliczone zostały podczas testowania gotowego modelu po jego nauczaniu i przedstawiają rzeczywiste wartości do weryfikacji na zbiorze testowym, składającym się z próbek reprezentujących każdą z 6 grup dla 3 analizowanych stali. Model z tym wynikiem został zaakceptowany i przygotowany do implementacji w całościowej strukturze sieci neuronowej.

MAE	MSE	R2
0.0478	0.0791	0.8898

Tabela 3.5: Metryki wydajności modelu MLP.

Sieć neuronowa oparta na podejściu *Curve to Curve* W tym innym podejściu do wcześniej opisanej sieci *Point to Point* postawiono na głębsze modele osadzone w bardziej rozbudowanej architekturze, co było wymagane przez alternatywne podejście, w którym nie generowano równolegle wielu punktów, składających się ostatecznie na zestaw danych, a bezpośrednio całą krzywą materiałową. Architektura sieci nakierowana jest na wychwytywanie i odtwarzanie złożoności krzywych materiałowych. Jako dane wejściowe wykorzystuje 5 kanałów: 2 pochodzące z krzywej, tj. siła i przemieszczenie, oraz 3 jako wektory materiału, aby wyspecjalizować model w każdym z materiałów z osobna. Model sieci zastosowanej tutaj to 1D CNN (z ang. *Convolutional Neural Network*), który został rozwinięty, aby móc generować ostatecznie krzywą naprężenie-odkształcenie dla każdego z badanych materiałów, które względem siebie przedstawiają unikalne zachowanie. Nauczanie rozpoczyna się od załadowania pakietu danych z przeskalowanej bazy danych SQLite, składającej się z kompletnych par krzywych w formacie JSON. Aby zapewnić unikalność każdej krzywej, zastosowano klasyfikację ich pod względem unikalnych identyfikatorów, przez co dana krzywa, niezależnie od liczby par, które utworzy, będzie znajdować się tylko w jednym zestawie danych: do uczenia, do walidacji lub do testu, unikając przeuczania oraz wycieku danych. Wielościowy podział na zbiory został przedstawiony w tabeli 3.6 i sumarycznie stanowi 52% łącznej sumy par krzywych w bazie danych. Aby zoptymalizować parametry, skrypt ponownie wykorzystuje bibliotekę Optuna. W tej konfiguracji jako cel wybrano 50 epok, z zastosowaniem mechanizmu cierpliwości uczenia, skutkującego zatrzymaniem procesu w przypadku braku poprawy strat walidacyjnych po 10 epokach. Po takim procesie najlepszy model zostaje zapisany A.11. Skrypt składa się z 3 głównych członów:

- Definicji klasy zbioru danych *SPTtoUTTDataset()* opisująca podział na zbiory danych do uczenia, walidacji i testowania oraz zapewniająca unikalność krzywych w każdym z zestawów.
- Definicji klasy modelu *CurveToCurveCNN()*, stwarzającej architekturę na której operacje wykonywał będzie skrypt
- Funkcji *objective()* zawierającej parametry do optymalizacji oraz schemat uczenia
- Funkcje pomocnicze, takie jak raportowanie czasu trwania procesu oraz wyników na żywo, a także generowanie raportów końcowych, są pomocne w ocenie wyników.

W ramach optymalizacji hiperparametrów najpierw podjęto próby ręcznego doboru wartości na podstawie danych sugerowanych w literaturze [39], a następnie, bazując na odkrytych trendach, dobrano zakresy hiperparametrów do optymalizacji z wykorzystaniem biblioteki Optuna. Poszukiwania przeprowadzono w dwóch etapach. Pierwszym szerszym, a następnie kolejnym, węższym, nakierowanym na dobór jak najlepszych wartości parametrów. Proces ten podzielono z uwagi na zaobserwowane podczas pierwszych prób uczenia długie czasy przejścia przez kolejne epoki, wynikające ze złożoności architektury. Proces nauczania maszynowego zakończył się sukcesem, a w jego wyniku wybrano następujące parametry:

- Wielkość warstwy ukrytej:
 - Zakres: 64, 128, 256, 384, 512
 - Wartość końcowa: 512

- Współczynnik *dropout*:
 - Zakres: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6
 - Wartość końcowa: 0.4
- Wielkość pakietów danych:
 - Zakres: 8, 16
 - Wartość końcowa: 16
- Funkcja aktywacji:
 - Zakres: *ReLU* lub *LeakyReLU*
 - Wartość końcowa: *ReLU* - funkcja lepiej współgra z wysokim współczynnikiem *dropout*

Jako wartości stałe przyjęto:

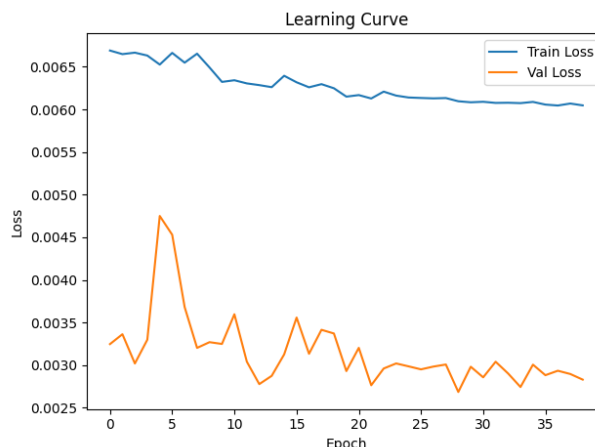
- Algorytm optymalizacji: AdamW - sprawdzający się lepiej w głębokich modelach [113]
- Prędkość uczenia: $1e-03$ - ze względu na plan zastosowania adaptacyjnej wartości w kolejnym etapie
- Rozmiar jądra (z ang. *Kernel size*): 7 - aby wykrywać globalne cechy krzywych materiałowych [39].

Dataset	Pairs	Unique SPT IDs	Unique UTT IDs
Train	3006	159	113
Val	72	24	18
Test	297	49	36

Tabela 3.6: Tabela podziału na zbiory danych do uczenia, walidacji i testów.

Tak przygotowane parametry architektury modelu sieci neuronowej zostały dalej wprowadzone w kolejny skrypt A.12 nakierowany na dalsze uczenie modelu, celem uzyskania jak najlepszych wyników straty walidacji. Jest to pełnoskalowe nauczanie sieci, polegające na zastosowaniu wolniejszego tempa nauczania oraz większej liczby epok, co pozwala modelowi przezwyciężyć momenty, w których straty walidacyjne nie spadają.

Logika zastosowana w dalszym uczeniu bazuje na warunkach stosowanych przy nauczaniu z biblioteką Optuna i rozpoczyna uczenie w momencie, w którym zakończyło się ono w poprzednim skrypcie. Wstępnie nauczony plik modelu zostaje załadowany do programu, a algorytm optymalizacji zakłada tempo uczenia = 0.0001, co jest wartością o rząd wielkości niższą niż w przypadku wstępnego uczenia maszynowego. Dodatkowym parametrem, który jest wprowadzony, jest rozkład wagi (z ang. *weight decay*) = 0,0005, wykorzystywany w przypadku algorytmu optymalizacji AdamW [113, 80]. Parametr ten zapewnia pewniejsze uczenie, bez ryzyka skupiania się modelu jedynie na parametrach o wysokich wartościach. Parametry te mają na celu poprawę zbieżności podczas uczenia, co dodatkowo zapewnia funkcja *ReduceLROnPlateau*, z cierpliwością 5 epok. W przypadku, gdy wyniki walidacji wypłaszczają się bez wyraźnej poprawy, skrypt



Rysunek 3.20: Wykres uczenia i walidacji na modelu 1D CNN

zmniejsza tempo uczenia o połowę, pozwalając skupić się na detalach. Ogranicza to ryzyko przeuczenia i zapewnia wysoką precyzję.

Ponownie, jak w przypadku uczenia maszynowego, pakiety danych zawierają unikalne krzywe. Identyfikatory krzywych do uczenia, jak i te do testowania, są zapisywane w plikach CSV. Zapewnia to odtwarzalność oraz możliwość weryfikacji poprawności i dokładności predykcji dla pełnej sieci neuronowej. Uczenie trwa docelowo 200 epok i jest to proces czasochłonny, w którym, w przypadku modelu tak głębokiego, trwa nawet kilkadziesiąt godzin. Aby zapewnić elastyczność operatorowi, każda poprawa w treningu nadpisuje najlepszy model, co zapewnia możliwość przerwania procesu.

Po zakończeniu uczenia model jest oceniany na podstawie zestawu testowego. Dla każdej próbki wyznaczone są trzy metryki: MSE (błąd średniokwadratowy), MAE (średni błąd bezwzględny) oraz współczynnik regresji R^2 . Metryki te są wyznaczone globalnie i dla każdego materiału. Wyniki są w plikach CSV. Skrypt w tej formie przedstawia ostateczny moment rozwoju modelu w ścieżce sieci neuronowej. Wyniki uczenia i walidacji modelu przedstawiono na wykresie 3.20, gdzie docelowa krzywa uczenia stale wykazuje tendencję spadkową, jednak krzywa walidacji jest poszarpana i dopiero na przestrzeni wielu epok wskazuje na uśredniony spadek. Podobnie jak w przypadku modelu MLP, wyniki weryfikacji parametrycznej przedstawiono w tabelicy 3.7, jednak w tym wypadku z podziałem na każdy z materiałów z osobna, a całkowite wyniki podsumowane są wartościami uśrednionymi. Widoczny jest niższy wskaźnik R^2 dla materiału M3, jednak zostanie on zweryfikowany dalej na pełnych krzywych materiałowych. Mimo jego zaniżenia, całkowity wynik modelu CNN jest o 4 punkty procentowe wyższy niż wynik modelu MLP. Model CNN został tym samym zaklasyfikowany do implementacji w programie automatyzacji oceny stanu materiału.

3.5.4 Automatyzacja oceny stanu materiału

Przygotowane wcześniej finalne modele sieci neuronowej zostały przygotowane do zaimplementowania w celu automatyzacji oceny stanu materiału. Architektura, w formie skryptu, była budowana etapowo. Pierwszym z etapów było sprawdzenie, czy funkcjonujący model jest zdolny do wygenerowania faktycznych danych, które posłużą do interpretacji wyników uzyskanych za pomocą SPT. Zadanie, jakim jest automatyzacja oceny stanu materiału, zakłada, że po otrzymaniu danych SPT w formie pliku tekstowego, będącego

Material	MAE	MSE	R2
D1	0.0513	0.0038	0.9422
H1	0.0224	0.0010	0.9535
J1	0.0437	0.0028	0.9692
M3	0.0866	0.0155	0.7803
R1	0.0186	0.0006	0.9543
T1	0.0186	0.0007	0.9516
AVERAGE	0.0402	0.0041	0.9252

Tabela 3.7: Metryki zgodności modelu CNN dla danego materiału.

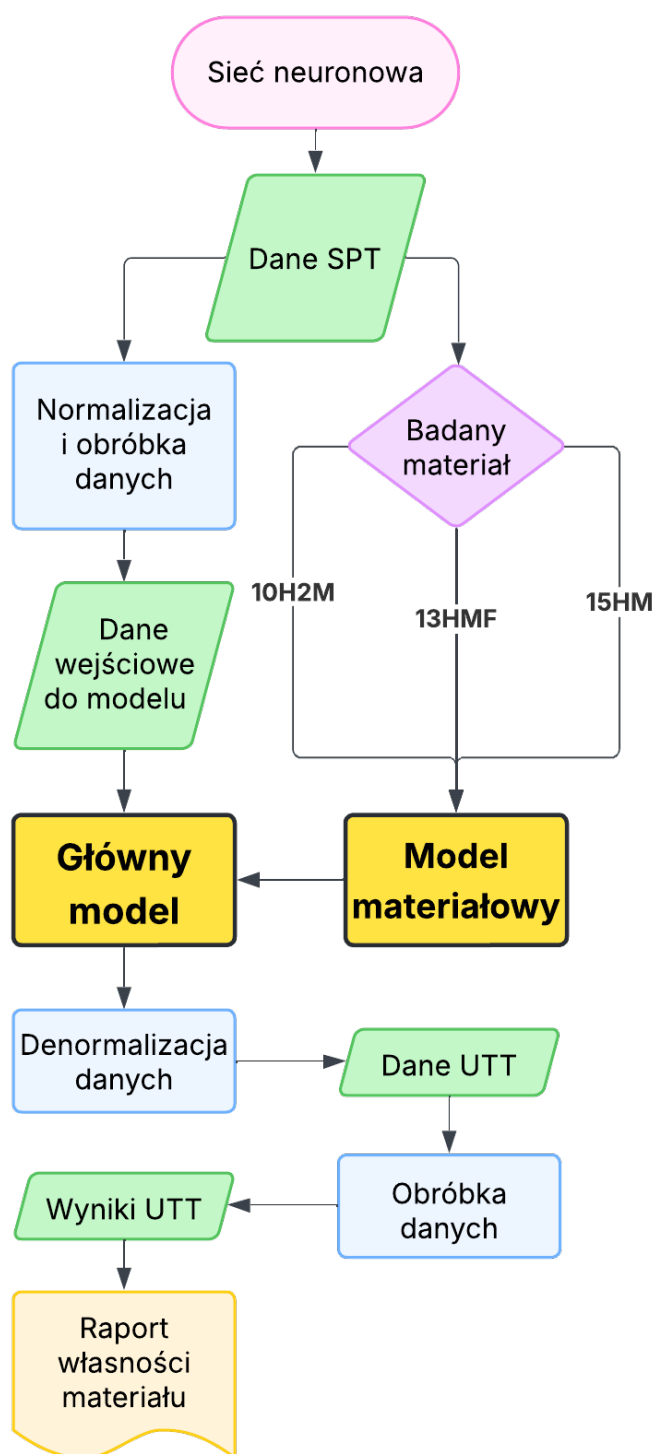
wynikiem testu SPT na maszynie wytrzymałościowej, program wykorzystujący model sieci neuronowej będzie w stanie podać parametry materiałowe. Działanie programu składa się ostatecznie z kilku głównych kroków, które w pętli są wykonywane dla wszystkich załadowanych do bazy danych. Dane wejściowe są podawane w formie surowej, wygenerowanej w fazie eksperymentu. Poszczególne etapy tej procedury przedstawiono na schemacie blokowym (Rysunek 3.21).

Każdy z etapów na początku pracy realizowany był w osobnych krokach, aby z sukcesem budować poprawne rozwiązania i poprawnie je walidować. W ramach pracy wszystkie etapy zostały ostatecznie scalone w jeden program usprawniający proces oceny parametrów materiałowych.

W pierwszym etapie zaimplementowano podejścia, jakie wykorzystywano do budowy bazy danych. Sieć jest przygotowana do pracy na określonym typie danych, dlatego zadaniem całego programu jest wstępne przetworzenie tych danych. Parametry stosowane do normalizacji przyjmują przedstawione wcześniej wartości.

Kolejny krok polegał na wykorzystaniu modelu, zgodnie z zastosowaniem, do wygenerowania danych UTT. Dane generowane były w stanie znormalizowanym, dlatego też należało wykorzystać współczynniki normalizacji UTT, używane na etapie generowania bazy danych materiałowych, do procesu de-normalizacji wyników, a następnie zapisać otrzymane wartości w pliku tekstowym. Skrypt rozwijany był począwszy od pracy na jednym pliku, a następnie do pracy na całych folderach i zbiorach, aby zmaksymalizować wydajność generowania.

Dane wygenerowane i zapisane w plikach przedstawiały surowe dane UTT, które można już wykorzystywać do samodzielnej analizy wyników. Tym samym cel badań został zrealizowany, ponieważ otrzymano dane UTT na podstawie danych SPT w sposób automatyczny. W celu pełnej automatyzacji program rozwijano dalej, dodając kroki z ostatniego etapu, jakim jest obliczanie właściwości materiałowych na podstawie krzywych wygenerowanych dla testu UTT. Do tego podejścia przyjęto wartości bazowe, pochodzące z oryginalnych danych UTT dla długości bazy pomiarowej oraz pola powierzchni przekroju poprzecznego próbki. Dzięki tym założeniom dla wirtualnych wyników UTT można było dobrać warunki brzegowe potrzebne do obliczenia naprężenia oraz odkształcenia. Aby uwzględnić wpływ wariacji grubości próbek SPT, wpływający na wartości siły na wykresie, zastosowano współczynnik korekcji (3.8), gdzie *correction* jest wartością bezwymiarową, a *u* podawane jest w μm . Współczynnik ani sama grubość próbek nie były uwzględniane na etapie budowania bazy danych lub trenowania modelu. Zweryfikowano natomiast brak wpływu grubo-



Rysunek 3.21: Schemat blokowy pracy sieci neuronowej

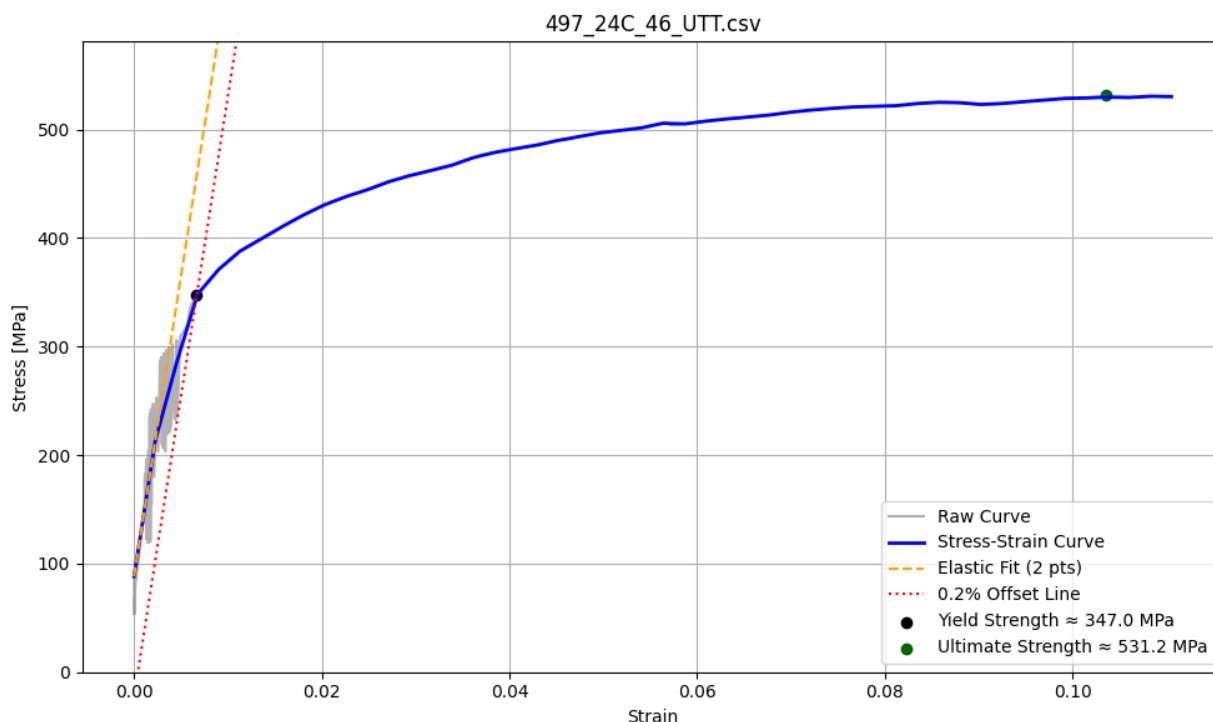
ści próbki na obliczone właściwości badanych materiałów. Ze względu na konstrukcję i parowanie próbek w bazie danych, gdzie przyjęto zasadę łączenia wszystkich próbek SPT w pary ze wszystkimi dostępnymi próbkami z UTT, naturalna wariacja wynikająca z różnorodności populacji zanika. Współczynnik zdecydowano się uwzględnić na etapie post-processingu danych opisywanych dalej w pracy, aby wprowadzić ponownie naturalną wariację wyników w zakresie dopuszczalnym przez naturalny rozkład wyników badanych próbek. Wyniki w formie surowej, nieprzetwarzane dalej po wygenerowaniu przez model, pozostają dostępne.

$$correction = \frac{500\mu m}{u} \quad (3.8)$$

Podczas wyznaczania granicy plastyczności kluczowe było, aby z danych wygenerowanych oraz przedstawionych na wykresie odizolować zakres sprężysty, gdzie wstępny zakres naprężenia i odkształcenia rośnie liniowo. Zadanie to zrealizowano poprzez wybór punktów o wartości odkształcenia mniejszej bądź równej 0.002. Jeśli w tym zakresie znajdują się punkty, obliczane jest równanie nachylenia z wykorzystaniem regresji liniowej, reprezentujące moduł Younga i obrazujące zakres sprężysty. Dzięki temu możliwe jest podjęcie próby wyznaczenia punktu na krzywej będącego momentem przejścia materiału w stan plastyczny.

W pracy przyjęto standardową metodę wyznaczania umownej granicy plastyczności R_{P02} , czyli wartości naprężenia, po przekroczeniu której materiał wchodzi w stan nieodwracalnych odkształceń. Na wykresie naprężenie-odkształcenie metoda ta polega na odczytywaniu wartości granicy plastyczności z pomocą prostej równoległej do zakresu sprężystego. Prostą równoległą wyznacza się poprzez odsunięcie o 0.2% lub 0.002 wartości bezwzględnej odkształcenia na osi X względem prostej reprezentującej zakres sprężysty. Wartość granicy plastyczności równa się wartości naprężenia w punkcie, lub najbliższej wartości temu punktowi, przecięcia tej prostej z wykresem naprężenie-odkształcenie. Tak wyznaczona granica plastyczności jest zapisywana, a program przechodzi do obliczania granicy wytrzymałości materiału, która w przypadku krzywej materiałowej z testu UTT przedstawionej w formie wykresu naprężenie-odkształcenie jest wartością najwyższego zarejestrowanego naprężenia. W przypadku krzywej siła-przemieszczenie obliczana jest poprzez stosunek najwyższej wartości wygenerowanej siły F_m w danych UTT do przekroju początkowego próbki S_0 przyjętego jako warunek brzegowy.

Należy nadmienić, iż otrzymane wykresy, niezależnie od architektury sieci, obarczone są błędami w postaci zaburzeń widocznych zwłaszcza we wczesnych krokach, w obszarze zakresu sprężystego, co nie stanowi przeszkody w interpretacji wykresu przez operatora, jednak stanowi znaczące utrudnienie dla automatyzacji predykcji i może powodować błędy w odczycie, konieczne do ręcznej korekty w przypadku praktycznych zastosowań. Aby jednak w trakcie procesowania danych nie usuwać rzeczywistych wartości z predykcji, np. poprzez dodanie funkcji monotonicznego wzrostu wartości, zdecydowano o zachowaniu wykresów w formie, w jakiej są generowane, a jedynie nakładaniu odpowiednich masek, aby ułatwić zadanie dla algorytmu. Przykładowy wykres zawierający surową krzywą wynikową, wygładzoną krzywą do predykcji, linię zakresu sprężystego oraz linię równoległą do zakresu sprężystego, odsuniętą o 0.2% odkształcenia. Na wykresie odczytano 2 punkty charakterystyczne dla badanego materiału przedstawiające granice wytrzymałości i granice plastyczności. Wykres nie zawiera zakresu szyjkowania materiału. Jako dodatkowe



Rysunek 3.22: Przykładowy wykres wygenerowany przez sieć neuronową

funkcje założono informowanie użytkownika o przetworzonych plikach oraz zapis wyników z całej grupy materiałów znajdujących się w folderze danych wejściowych, aby można było podjąć pierwsze statystyczne analizy. Każdy skrypt sieci neuronowej, niezależnie od zastosowanej architektury, umożliwia dodanie dodatkowych funkcji, jednak w aktualnej raportowanej formie stanowi on spełnienie założeń niniejszej pracy.

Sieć neuronowa Skrypt A.6 opisujący całkowity proces automatyzacji oceny stanu materiału z wykorzystaniem danych SPT oraz modelu MLP składa się z:

- Definicji klasy modelu *MaterialAwareNeuralNetwork()*
- Funkcji normalizacji danych *normalize_data*
- Funkcji uzyskiwania od użytkownika informacji o badanym materiale *get_user_input()*
- Funkcji denormalizacji danych wyjściowych i generowania danych UTT *generate_utt_data*
- Funkcji generowania krzywych naprężenie - odkształcenie i wyznaczania własności materiału *draw_graphs_and_calculate_properties*
- Funkcji zapętlających *process_all_steps* oraz *process_all_files* aktywujących wszystkie pozostałe kroki dla wszystkich plików.

W przypadku sieci neuronowej opartej na architekturze CNN skrypt składa się z zasad podobnych funkcji, tj. denormalizacji, generowania plików UTT, uzyskiwania informacji od operatora oraz funkcji zapętlających, jednak rozbudowanych o dodatkowe moduły. Główny skrypt A.13 ponadto składa się z dodatkowej

funkcji *adaptive_sample_curve* adaptacyjnego skalowania danych wejściowych SPT, składających się z tysięcy punktów, do zbioru 300 punktów odpowiadających tej samej funkcji używanej przy tworzeniu bazy danych.

Generowanie wykresów i odczyt własności materiału Każda z sieci, w celu wygenerowania gotowych wykresów oraz raportów z obliczonych własności, korzysta z funkcji *analyze_and_plot_all* w sposób jednakowy, niezależnie od modelu. Jako człon zaciągany do głównych programów sieci neuronowych, skrypt A.14 ma specyficzne funkcje, nakierowane na wygenerowanie danych gotowych do oceny przez operatora. Ze względu na występujące zakłócenia w danych generowanych oraz potrzebę dostosowania metod wyznaczania własności materiału do generowanych danych, zdecydowano się dodać dodatkową maskę pseudo-wygładzającą, nakładaną na 300-punktową krzywą w przypadku modelu CNN oraz na pełną krzywą o skali ponad 10 000 punktów dla modelu MLP, która dalej skaluje je równomiernie do 50 punktów. Dzięki temu możliwe jest automatyczne odczytanie tych wartości z krzywej, które operator zweryfikuje samodzielnie, patrząc na gotowy, wygenerowany wykres. Sam skrypt dzieli się na trzy funkcje:

- Główna funkcja *analyze_and_plot_all* zapętlająca, stanowiąca jądro programu, odpowiada za skalowanie danych wejściowych, ładowanie funkcji pomocniczych oraz generowanie danych wyjściowych w postaci raportu i wykresów.
- Funkcji *elastic_fit_by_two_points* do wyznaczania spodziewanego zakresu sprężystego na wykresie
- Funkcji *calculate_Re_offse* do wyznaczania prostej równoległej do zakresu sprężystego i odsuniętej zgodnie z przyjętym standardem o 0.2% odkształcenia.

Klasyfikator SPT Jako dodatkowy program możliwy do implementacji w każdej z sieci neuronowych przygotowano klasyfikator przyporządkowujący dane wejściowe SPT tuż po normalizacji do jednego z dwóch stanów obecnie znanych dla materiału, np. dla stali 10H2M, klasyfikator ocenia, czy krzywa SPT pochodzi z grupy D1 - stan nowy stali 10H2M, czy też z grupy J1 - materiał po 300000 godzin pracy. Klasyfikator (Skrypt A.15) przygotowany został na podstawie prostego modelu sieci neuronowej 3-warstwowej typu CNN. Opiera się on na bazie danych typu *Curve to Curve*, a przy nauczaniu zastosowano tę samą logikę, jak w przypadku pełnego modelu CNN tj. unikalność krzywych pomiędzy zbiorami testowym, walidacyjnym i uczącym. Jako warunki końcowe przy procesie uczenia przyjęto maksymalnie 500 epok, poprawność predykcji podczas uczenia min. 95% oraz aby zagwarantować, że model jest naprawdę nauczony, a nie jedynie losowo poprawnie przewidział wynik, dodano warunek spadku średniej straty walidacji przed zakończeniem uczenia do minimum 0.05. Po zakończeniu uczenia model został przetestowany, a wyniki zaprezentowano w tabeli 3.8. Wartość NOK oznacza, że klasyfikator dla przedstawionej próbki błędnie założył jej stan, np. dla stali 13HMF, grupy M3 jedna z próbek została oceniona jako przynależna do grupy R1 tj. materiału przepracowanego.

Batch	Total Samples	NOK Count	NOK File
D1	47	1	503_25C_42.txt
J1	46	0	—
M3	38	1	500_22C_221.txt
R1	32	0	—
T1	33	1	488_22C_33.txt
H1	35	0	—

Tabela 3.8: Wyniki testów klasyfikatora materiału

3.5.5 Automatyzacja analizy statystycznej

Celem usprawnienia wykonywania analizy statystycznej w niniejszej pracy oraz na poczet wszelkich przyszłych badań zdecydowano się na częściowe przejście z komercyjnych narzędzi statystycznych do skryptów przygotowanych specjalnie do zadania, jakim jest analiza porównawcza wyników z sieci neuronowej (NN) w porównaniu do wyników UTT i SPT. Efektem tych działań było przygotowanie skryptów w języku Python w dwóch wariantach: do analizy SPT i UTT A.7 oraz do pełnej analizy SPT, UTT i NN A.8. Funkcjonalność programu polega na automatycznym przygotowaniu danych z plików w formacie .xlsx, zawierających zebrane dane pochodzące z różnych testów. Program odszukuje plik, sprawdza, czy zawiera on strony Re oraz Rm, a następnie weryfikuje, czy kolumny danych w pliku są zgodne z przyjętym założeniem.

Zadania programu pod kątem analizy dzielą się na:

- Usunięcie wyników statystycznie odstających od całości grupy z wykorzystaniem metody wyznaczania zmodyfikowanego wyniku Z
- Analiza dystrybucji wyników oraz wyznaczenie najlepszego dopasowania, jeśli rozkład nie spełnia warunków normalności
- Wyrysowanie histogramów wraz z funkcją dopasowania
- Przygotowanie wykresów prawdopodobieństwa z przedziałem ufności 95% jako wartość podstawowa
- Zestawienie histogramów oraz wykresów prawdopodobieństwa dla wszystkich analizowanych grup danych: UTT oraz SPT, a dalej, przy analizie wyników generowanych przez sieć neuronową UTT, SPT oraz NN
- Zestawienie wszystkich wyników analizy statystycznej w tabeli gotowej do porównania, zawierającej:
 - Badaną własność materiału (Re i Rm)
 - Grupę materiału (np. D1 UTT)
 - Wartość średnią danej własności w MPa
 - Medianę danej własności w MPa
 - Odchylenie standardowe w danej grupie

- Wartość parametru AD
- Wartość parametru p
- Ocena rozkładu
- Propozycje najlepszego rozkładu statystycznego
- Wartość N dla przeanalizowanej populacji
- Liczbę odrzuconych odstających wyników z danej grupy

Plik źródłowy z danymi jest nadpisywany zaktualizowaną wersją zawierającą nowe strony i tabele. Wykresy zapisywane są w folderze *output_analysis* z podziałem na podfoldery, każdy nazwany zgodnie z nazwą pliku źródłowego. Skrypt zapewnia, że do analizy brane są pod uwagę jedynie wyniki, które są statystycznie istotne, zapewniając automatyczną filtrację wyników odstających.

Rozdział 4

Wyniki badań

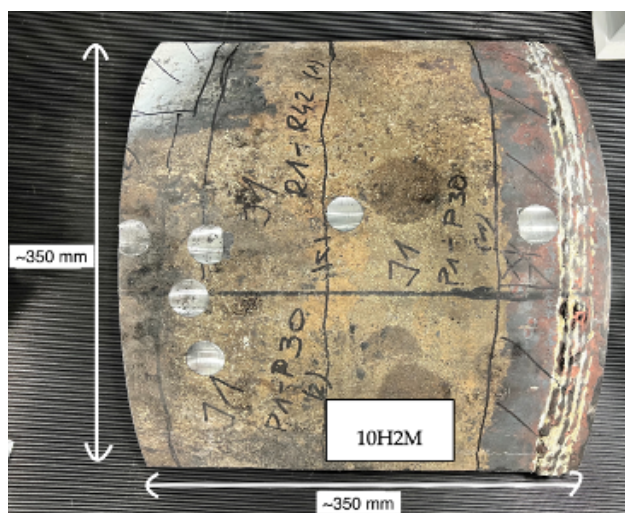
W niniejszym rozdziale ujęto i opisano zbiorcze wyniki badań eksperymentalnych dla całej puli łącznie ponad 400 wyników testów SPT oraz 180 wyników badania UTT dostępnych w laboratorium SPT. Przedstawiono również analizy statystyczne dla wszystkich metod eksperymentalnych i komputerowych oraz efektów modelowania z użyciem sieci neuronowych. Dla badań eksperymentalnych zwracając uwagę na różnice i podobieństwa w otrzymywanych wartościach. Dla modelowania z użyciem sieci neuronowych typu CNN oraz MLP w zestawieniu z wynikami UTT ujęto zalety, wady oraz ogólne charakterystyki sieci na podstawie otrzymanych wartości. Ze względu na ilość wygenerowanych danych, w pracy zaprezentowano jedynie przykładowe wyniki cząstkowe w formie graficznej, a całość wyników badań dostępna jest do wglądu w repozytorium [61].

4.1 Badania eksperymentalne i porównanie metod dla badanych materiałów

4.1.1 Stal 10H2M

SPT i UTT Zgodnie z przyjętym założeniem podjęto próbę statystycznego opisu wyników badań eksperymentalnych dla pierwszego badanego materiału, jakim jest stal 10H2M, została przeprowadzona kompleksowa analiza stanu materiału z wykorzystaniem metody SPT oraz UTT. Wykorzystano jednak dodatkowy wskaźnik pomocniczy, a zarazem weryfikujący, jakim była dostępna analiza mikrostruktury przeprowadzona przez laboratorium UDT. Z wykorzystaniem tak wielostronnego podejścia za cel obrano ocenę degradacji materiału pod kątem zdolności do dalszego wykorzystania w oryginalnej instalacji przemysłowej, z której po jej wyłączeniu z użytku został pobrany materiał do badań.

Materiał do badań zgodnie z przedstawionym w poprzednim rozdziale podziałem przygotowany był z dwóch różnych konstrukcji źródłowych. W stanie bazowym, niezmechanizowanym, o nazwie kodowej D1, oraz w stanie przepracowanym, o nazwie kodowej J1, pochodzącym z konstrukcji zbiornika ciśnieniowego. Oba warianty próbek zostały poddane testom SPT oraz UTT, aby zweryfikować przydatność i zdolności testu SP, a także aby rozpoznać podobieństwa lub różnice pomiędzy obydwoma metodami. Materiał D1, w trakcie pracy przez 250 tysięcy godzin, poddawany był stałemu ciśnieniu i warunkom pełzania przy temperaturach



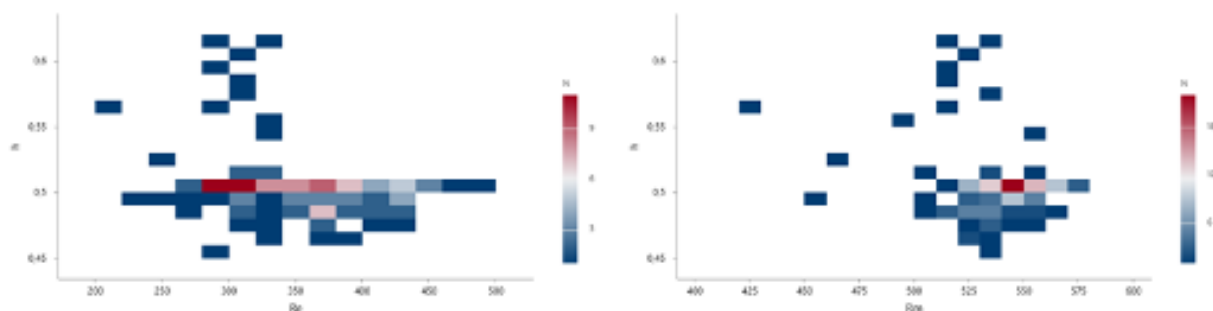
Rysunek 4.1: Materiał badawczy J1 - 10H2M w stanie przepracowanym.

do 500°C. Po tym okresie zbiornik został wyłączony z użytku i rozebrany, a próbki zostały pozyskane przez laboratorium SPT na cele badawcze. Materiał pobierany był z powierzchni zewnętrznej próbki, będącej wycinkiem zbiornika (Rysunek 4.1). Materiał pobierany był metodą Scooping, a łezki wycinane były z powierzchni zewnętrznej, zapewniając jedność mikrostrukturalną materiału dla wszystkich próbek. Dla badań UTT fragment konstrukcji odcięty został z tego samego fragmentu dostępnego w laboratorium i przekazany do laboratorium zewnętrznego do przygotowania próbek testowych do metody klasycznej.

Przed przystąpieniem do zasadniczych działań mających na celu analizę wyników zdecydowano się zweryfikować rzeczywisty wpływ grubości próbki SPT badanego materiału na wyniki. Celem tego zabiegu było poszerzenie tolerancji proponowanej przez normę opisującą metodę SPT. W tym celu zastosowano analizę z wykorzystaniem map ciepła. Mapy ciepła, przedstawione na Rysunku 4.2, przyczyniły się do lepszego zobrazowania uzyskanych obserwacji. Co istotne, nie wykazały one bezpośredniej, liniowej korelacji między zmianami grubości a wynikami testów. To kluczowe spostrzeżenie pozwoliło na rozszerzenie analizy na cały zakres badanych próbek, zwiększając tym samym użyteczność i zakres prowadzonych pomiarów. Efektywnie rozszerzono tolerancję grubości z ± 0.005 mm do ± 0.01 mm. Dzięki mapom cieplnym potwierdzającym przydatność wszystkich próbek, do dalszej analizy zakwalifikowano łącznie 198 próbek. Próbki pochodzące z testów UTT obejmowały 61 egzemplarzy, z czego 31 pochodziło z materiału D1, a 30 z materiału J1. W przypadku reprezentacji SPT przeanalizowano 137 próbek, z czego 68 pochodziło z materiału D1, a 69 z materiału J1.

Jako podstawowa ocena przydatności całej wynikowej analizy zastosowano klasyfikację danych odstających metodą zmodyfikowanego Z-score. W wyniku czego określono wartości odstające, niebiorące udziału w dalszej analizie. Wartości pozostałe opisano podstawowymi parametrami statystycznymi, wykonując operacje określające następujące wartości dla danej populacji danych:

- Średnia wyników
- Mediana wyników
- Odchylenie standardowe



Rysunek 4.2: Mapy ciepła, obrazujące zależność grubości próbki testowej z: a) Granicą plastyczności; b) Granicą wytrzymałości.

- Współczynnik AD
- p-value
- Hipotezę normalność rozkładu
- Najlepsze dopasowanie rozkładu
- Liczbę próbek poddanej finalnej analizie
- Liczbę danych odrzuconych w teście zmodyfikowanego Z-score

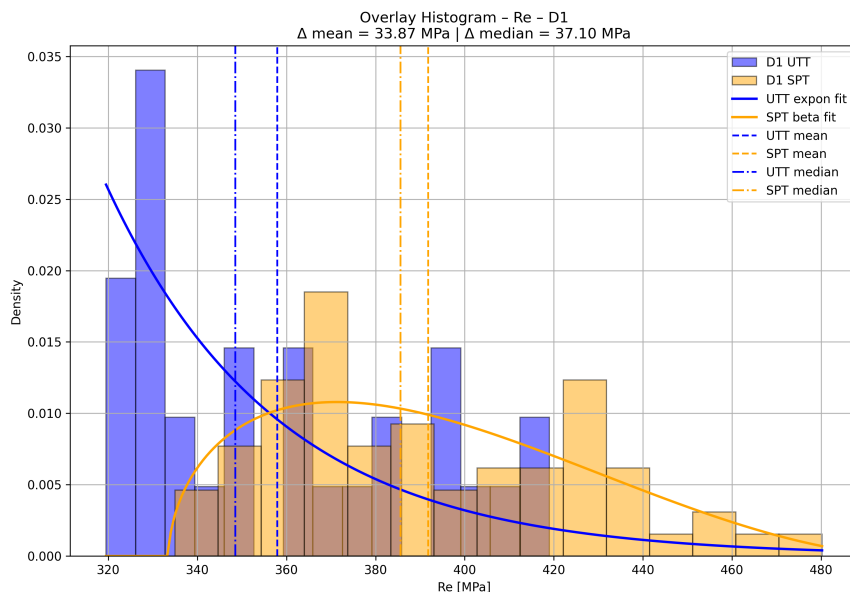
Zestawienie otrzymanych wartości zostało przedstawione w tabeli wyników (Tablica 4.1).

Property	Material	Mean [MPa]	[%] AVG Δ to UTT	Std Dev	AD Statistic	Shapiro p	Normality	Best-Fit Dist	N	Outliers Removed
Re	D1 SPT	392	9.5%	34.2	0.9888	0.0202	Not Normal	gamma	68	0
Re	D1 UTT	358	-	12	1.1796	0.0004	Not Normal	expon	31	0
Re	J1 SPT	311	-5.8%	26.2	0.8289	0.8326	Normal	norm	67	2
Re	J1 UTT	330	-	9.6	0.4429	0.6313	Normal	norm	29	1
Rm	D1 SPT	541	2.9%	14.7	0.7093	0.0925	Normal	norm	60	0
Rm	D1 UTT	526	-	11.6	0.8759	0.0289	Not Normal	weibull_min	31	0
Rm	J1 SPT	542	3.0%	15.7	1.4304	0.0004	Not Normal	weibull_min	65	4
Rm	J1 UTT	526	-	4.5	0.4755	0.3688	Normal	norm	30	0

Tabela 4.1: Wartości wyników badań metodami SPT i UTT dla materiału 10H2M.

Przedstawione wyniki dotyczące wytrzymałości na rozciąganie dowiodły, że metoda SPT była równie skuteczna w precyzyjnej ocenie tej wartości pomiarowej, jak można było tego oczekiwać w przypadku próby rozciągania (UTT). Chociaż uzyskane wyniki nie były identyczne, zaobserwowano pewną prawidłowość – nie wystąpiło przesunięcie wartości wytrzymałości na rozciąganie między poszczególnymi etapami eksploatacji materiału, a jedynie początkowa wartość uzyskana metodą SPT została przesunięta o 15 MPa w górę porównując wartości średnie obydwu grup.

W przypadku wartości granicy plastyczności nie zaobserwowano jednoznacznej korelacji pomiędzy obiema metodami. Zwrócono jednak uwagę na fakt, że metoda SPT wykazała większą czułość, co przejawiało się większym przesunięciem parametrów materiałowych oraz dokładniejszym wykrywaniem progu plastyczności w porównaniu z metodą tradycyjną.



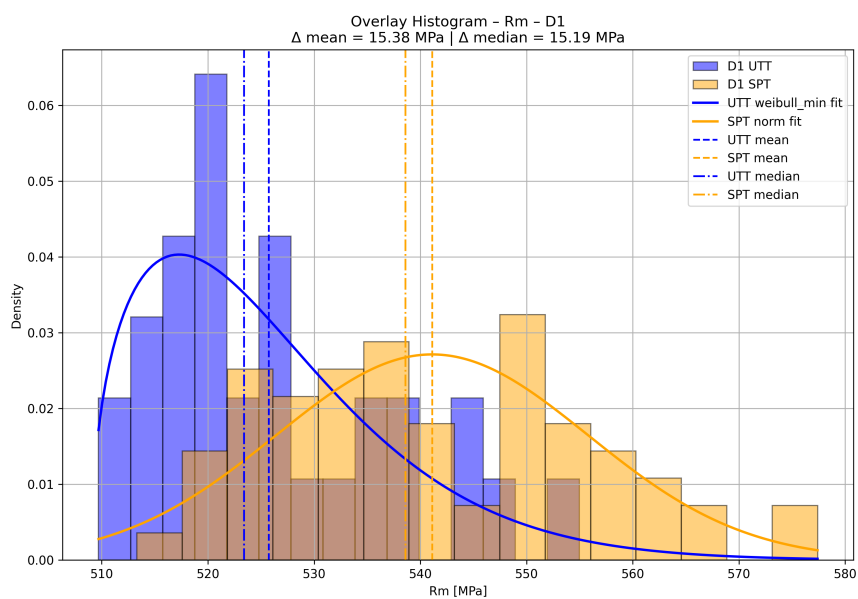
Rysunek 4.3: Histogramy wyników D1 - Granica plastyczności

W obu przypadkach metoda SPT dostarczyła informacji o stanie materiału w sposób analogiczny do UTT, jednak umożliwiła jej zastosowanie w eksploatowanych urządzeniach. Na przedstawionych histogramach (Rysunki 4.3, 4.4, 4.5, 4.6) zaobserwowano, że odchylenie wartości granicy plastyczności dla SPT pozostawało wysokie, co oznaczało, że dokładna ocena tej wartości była możliwa jedynie po przeprowadzeniu wielu testów na próbkach wyekstrahowanych z oryginalnego materiału.

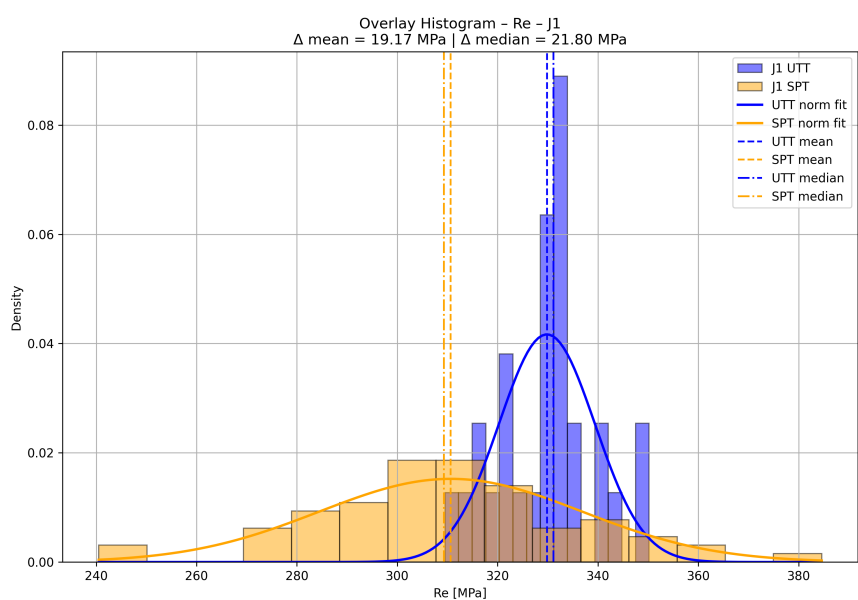
Przyjmując za punkt odniesienia wartości przedstawione w tabeli 3.3 dla stali 10H2M poddanej obróbce cieplnej, stwierdzono, że ani granica plastyczności, ani wytrzymałość na rozciąganie badanej próbki w stanie J1 nie osiągnęły jeszcze granicy eksploatacyjnej materiału. Oczekiwany spadek granicy plastyczności był zauważalny, co wynikało z długotrwałego oddziaływania naprężeń i temperatury, jednak pozostawał on wciąż na poziomie umożliwiającym dalsze użytkowanie materiału w warunkach eksploatacyjnych. Twierdzenie to zweryfikowano dalej poprzez analizę mikrostrukturalną.

Aby zbadać rozkład danych dla wszystkich grup pomiarowych na podstawie obliczonych parametrów, wygenerowano wykresy prawdopodobieństwa dla rozkładu normalnego, przedstawione na Rysunku 4.8 i Rysunku 4.7. Dopasowanie punktów danych do linii referencyjnej, w połączeniu z wartościami testu Andersona-Darlinga, wskazuje na przeważnie normalny rozkład wytrzymałości materiału, przy czym wartości średnie i odchylenia standardowe charakteryzują tendencję centralną oraz zmienność danych. Należy tu zwrócić uwagę na rysunku 4.7 jak silnie zarysowany jest podział wyników pomiędzy materiałem D1 i J1 dla testu SPT, i jak kontrastuje to z niemalże zlewającymi się wynikami dla UTT. Badanie SPT wskazuje na silną czułość aplikującą subtelne zmiany zaistniałe w materiale po eksploatacji, które są niewidoczne dla testu UTT, na co wskazują otrzymane wyniki.

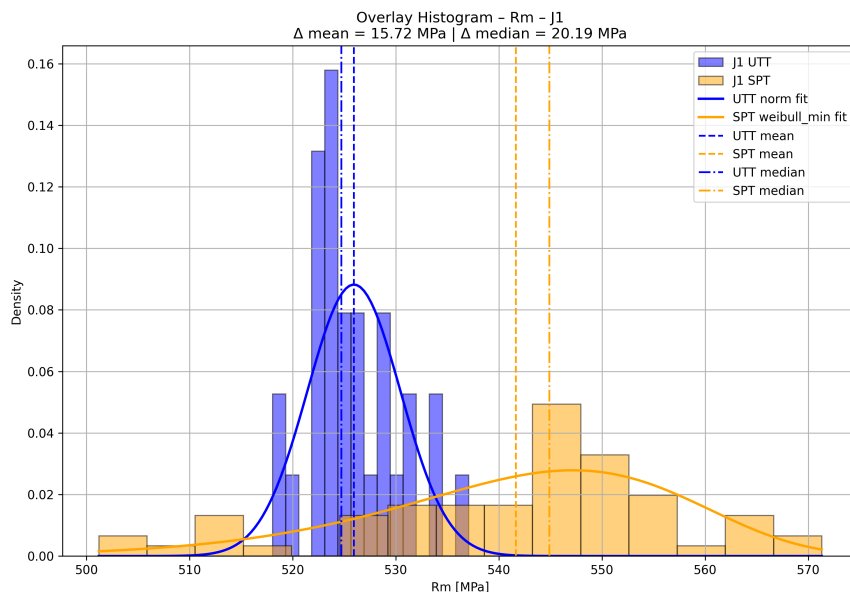
Liczebność próbek (N) była wystarczająco duża, aby zapewnić istotność statystyczną, z drobnymi wyjątkami, gdzie wartości p sugerowały odchylenia od normalności. Analiza porównawcza grup testowych ujawniła spójne wzorce rozkładu, przy czym większość zbiorów danych ściśle przybliżała oczekiwaną linię rozkładu normalnego, co przedstawiono w kolumnie 8 tabeli 4.1. Spośród wszystkich analizowanych



Rysunek 4.4: Histogramy wyników D1 - Granica wytrzymałości



Rysunek 4.5: Histogramy wyników J1 - Granica plastyczności



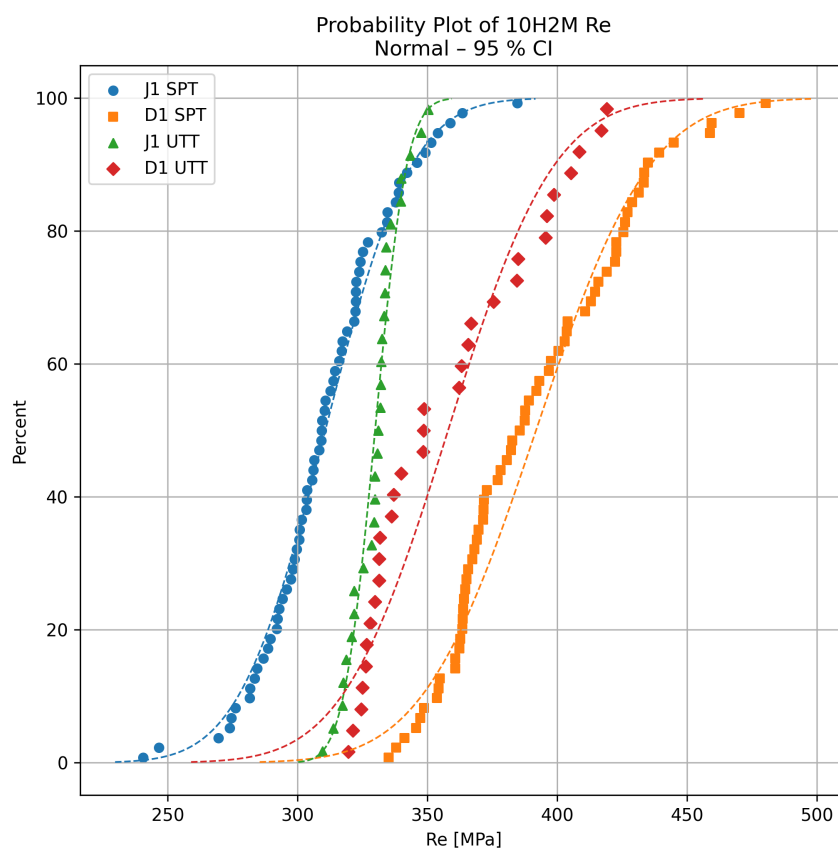
Rysunek 4.6: Histogramy wyników J1 - Granica wytrzymałości

grup część nie pasowała do kategorii rozkładu normalnego. Jest to zjawisko naturalne w przypadku badania materiałów, gdzie próbki pobierane były z różnych obszarów oraz w przypadku naturalnej wariacji wynikającej z metody pomiarowej. Wszystkie wykresy zostały zdeponowane w repozytorium do niniejszej pracy [61]. Oznacza to, że wyniki uzyskane w analizie charakteryzują się skrzywionym rozkładem, z koncentracją niższych wartości i ogonem obejmującym rzadsze, wyższe wartości naprężeń. Mierzone właściwości materiału podlegają losowej zmienności, a nie systematycznym błędom. Zapewnia to solidne podstawy do statystycznej kontroli jakości w procesie badania materiałów.

W kolejnych krokach wygenerowano dendrogram dla wyników SPT, który w analizach klasteryzacji hierarchicznej stanowi graficzne odwzorowanie relacji między punktami danych oraz organizacją skupień na różnych poziomach podobieństwa. Dendrogramy ułatwiają zrozumienie hierarchii skupień i dostarczają cennych informacji dotyczących optymalnej liczby klastrow.

Struktura dendrogramu składa się z liści, gałęzi i węzłów. Każdy liść symbolizuje pojedynczy punkt danych, natomiast gałęzie łączą liście i węzły, przedstawiając proces łączenia klastrow. Węzły oznaczają miejsca, w których gałęzie się łączą, co sygnalizuje tworzenie klastrow wyższego rzędu poprzez scalanie klastrow niższego rzędu. Wysokość gałęzi w dendrogramie odzwierciedla stopień niepodobieństwa między łączonymi klastrami, gdzie zauważono, że im większa wysokość, tym większa różnica między klastrami. Wysokości te są obliczane na podstawie wybranej miary odległości oraz zastosowanej metody łączenia. Pierwszym podejściem było zastosowanie analizy połączonej dla całej populacji próbek pochodzących ze wszystkich źródeł i w obu stanach w celu sprawdzenia, na ile są one od siebie rozróżnialne (Rysunek 4.9).

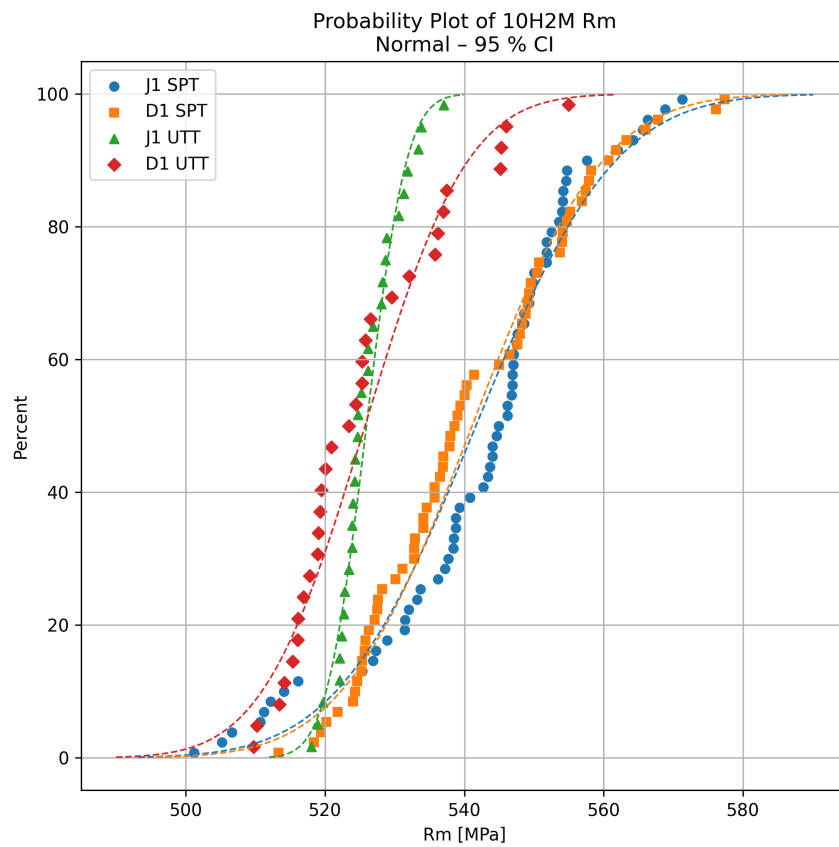
Łącznie utworzono sześć klastrow, wyodrębnionych z czterech oddzielnych grup pomiarowych, obejmujących dwie metody badawcze oraz dwa stany materiału. Jedynie klastry 1 i 2 oraz ich centroidy odzwierciedlają rzeczywiste zmierzone wartości granicy plastyczności i wytrzymałości na rozciąganie stali 10H2M (Tabela 4.2). Próbki z klastra 5 w większości pochodzą z grupy D1 SPT oraz kilku próbek z D1 UTT. Wyniki te stanowią mieszankę wyników UTT i SPT, co potwierdza ich użyteczność w praktyce poprzez wzajemne



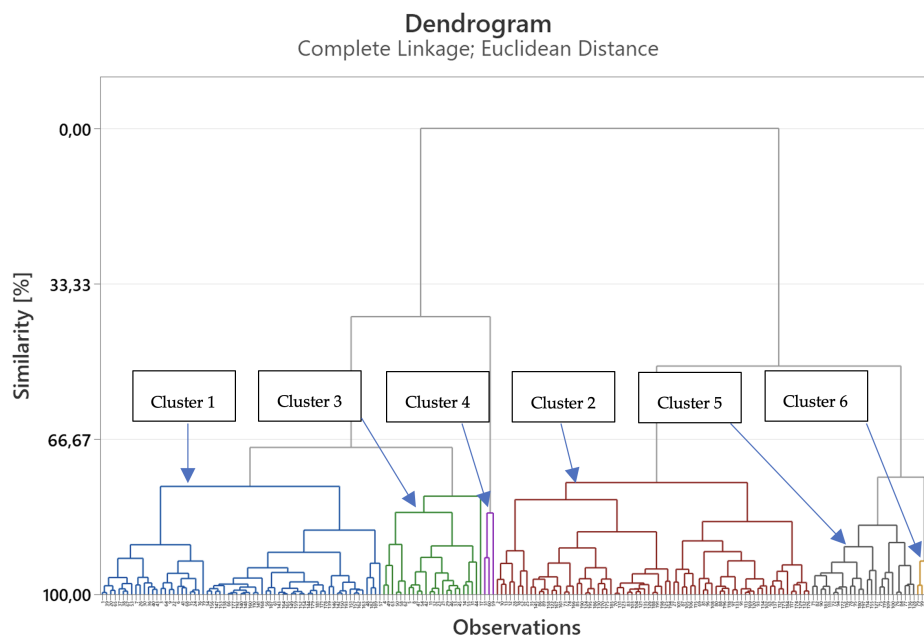
Rysunek 4.7: Wykresy prawdopodobieństwa rozkładu dla stali 10H2M - Granica plastyczności

Variable	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Grand cent.
Rm	532.40	534.04	531.80	448.43	553.31	561.80	534.91
Re	322.03	365.38	287.75	221.47	421.93	467.13	348.32
N	67	75	24	3	25	4	198

Tabela 4.2: Wyniki analizy klastrow dla całej populacji próbek 10H2M



Rysunek 4.8: Wykresy prawdopodobieństwa rozkładu dla stali 10H2M - Granica wytrzymałości



Rysunek 4.9: Dendrogram z analizy klastrow dla materiału 10H2M

uzupełnianie się w celu uzyskania szerszego zakresu analizy mechanicznej. Klastry 3 do 6 obejmują wyniki testów o dużych odchyleniach od przewidywanych wartości naprężeń oraz wyniki znajdujące się poza zakresem skali zarówno w jej dolnym, jak i górnym przedziale.

Variable	Cluster 1	Cluster 2	Grand Centroid
Rm	534.16	543.81	538.95
Re	306.26	393.36	349.50
N	69	68	137

Tabela 4.3: Centroidy dla wszystkich próbek SPT materiału 10H2M

Podział próbek na dwa odrębne klastry metodą K-średnich dla wyników SPT, obejmujących zarówno próbki dobrej, jak i złej jakości, skutkował równomiernym podziałem całej początkowej populacji 137 próbek (Tabela 4.3). Ostatecznie do klastra 1 przypisano 69 próbek, a do klastra 2 – 68 próbek. Centroidy obu klastrów poprawnie odwzorowały wartości mechaniczne materiałów D1 i J2, uzyskane w klasycznej analizie danych przeprowadzonej na początku badań. Całkowita zgodność centroidów z oryginalnymi wartościami wyniosła 91%, co potwierdza ich trafność w identyfikacji rzeczywistych właściwości mechanicznych badanych próbek.

Podobne obserwacje poczyniono w przypadku podziału próbek SPT na cztery kategorie, obejmujące nie tylko materiały J1 i D1, ale także próbki niższej jakości, w celu określenia ich przynależności do głównych grup. Wyniki przedstawione w Tabeli 4.4 wskazują, że klastry 2 i 3 składają się z próbek spoza standardowego zakresu i mogą zostać sklasyfikowane jako błędy pomiarowe oraz próbki nieprzydatne do analizy. Ogólne podobieństwo członków klastrów do ich pierwotnych źródeł wynosi 55%, jednak wartość ta nie może być przyjęta bez dalszej dyskusji, ponieważ wiele próbek pochodzących z partii o niskiej jakości zostało zaklasyfikowanych jako dobre reprezentacje wartości właściwości mechanicznych, co zakłóca korelację.

Tabela 4.4: Centroidy dla 4 grup próbek SPT materiału 10H2M

Variable	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Grand Centroid
Rm	538.81	470.62	554.69	537.47	538.95
Re	306.06	236.10	427.47	365.33	349.50
N	54	5	27	51	137

Metoda K-średnich (K-Means) została ponownie zastosowana do wyników UTT w celu oceny, na ile możliwe jest ich rozróżnienie oraz jak wrażliwe są te metody w porównaniu do SPT. Wyniki przedstawione w Tabeli 4.5 pokazują nierównomierny rozkład wyników pomiędzy odpowiadającymi im grupami, przy czym większość próbek została przypisana do klastra 1, reprezentującego niższe właściwości mechaniczne. Ich zgodność z oryginalnymi źródłami wynosi około 70%, przy czym 39% próbek nowego materiału zostało zaklasyfikowanych jako próbki intensywnie eksploatowane. Wynika to z faktu, iż nowy materiał w swoim rozkładzie populacji zawierał wiele wyników granicy plastyczności, które nie różniły się od mate-

riału eksploatowanego, a przejawiał przy tym wyjątkowo wysokie odczyty dla kilku próbek, co skutkowało przyporządkowaniem ich do klastra drugiego oraz zgromadzeniem w tym podziale pozostałych nie tak ekstremalnych próbek. Wskazuje to na poprawny podział klastrów.

Variable	Cluster 1	Cluster 2	Grand centroid
Rm	523.44	534.65	525.83
Re	332.16	395.58	345.68
N	48	13	61

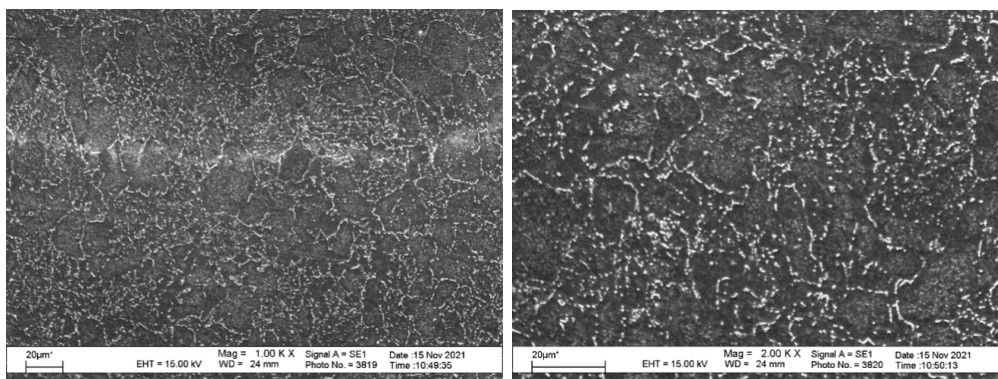
Tabela 4.5: Centroidy dla wszystkich próbek UTT materiału 10H2M

Analiza mikroskopowa Wtrącenia oraz puste przestrzenie mogą działać jako koncentratory naprężeń, prowadząc do powstawania mikropeknięć. SEM pozwala na identyfikację tych nieciągłości poprzez analizę ich rozmiaru, kształtu i rozmieszczenia w mikrostrukturze stali. Wtrącenia mogą przybierać różne morfologie, w tym kształt sferyczny, wydłużony lub nieregularny, i mogą być widoczne jako jasne lub ciemne punkty.

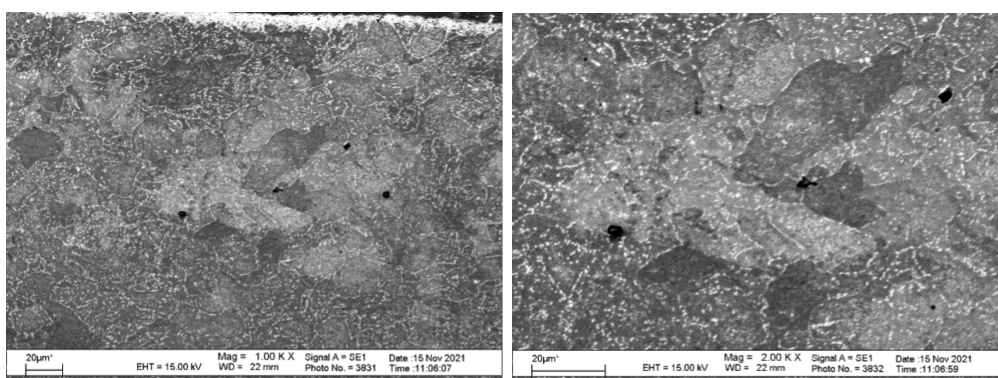
SEM umożliwia także identyfikację pęknięć wzdłuż granic ziaren, które mogą pojawiać się jako cienkie, ciemne linie oddzielające sąsiadujące ziarna. Pęknięcia te mogą być skutkiem naprężeń, korozji lub innych czynników. Obszary lokalnej deformacji w mikrostrukturze stali mogą wskazywać na obecność mikropeknięć lub innych nieciągłości. Deformacja ta może objawiać się jako pasma poślizgu, czyli płaskie struktury wewnątrz ziaren powstałe w wyniku ruchu dyslokacji. Pasma poślizgu mogą mieć formę równoległych linii lub pasm o zróżnicowanej szerokości i orientacji, a także mogą krzyżować się lub gromadzić wokół nieciągłości [36, 91].

Próbki pobrane z warstwy wewnętrznej (Rysunek 4.10), zewnętrznej (Rysunek 4.11) oraz środkowej (Rysunek 4.12) wykazują identyczne zachowanie, co pozwoliło na sformułowanie wspólnych wniosków dla wszystkich obszarów. Zaobserwowano strukturę ferrytyczno-bainityczną, która typowa jest dla stali niskowęglowych i o średniej zawartości węgla, szczególnie po obróbce termicznej stosowanej dla stali kotłowych. Sama struktura składa się z ferrytu oraz bainitu. Ferryt to faza żelaza α o strukturze sieci krystalograficznej przestrzennie centrowanej i stosunkowo niskiej twardości. Na zdjęciach mikroskopowych rozróżnialna jako jasne obszary. Bainit natomiast jest fazą powstałą podczas chłodzenia stali w zakresie temperatur pomiędzy dolnym zakresem perlitu a górnym zakresem powstawania fazy martenzytycznej. Sam bainit zawiera płytkowe lub igiełkowe wydzielения cementytu w osnowie ferrytycznej. Na zdjęciach widoczne są jego bardziej skondensowane obszary, co może prowadzić do anizotropii własności pomiędzy obszarami o mniejszym zagęszczeniu bainitu. Wynikać to może z lokalnych różnic w chłodzeniu podczas obróbki cieplnej i charakteryzuje się twardszymi i mniej plastycznymi obszarami otaczającymi ferryt. Wydzielения te obecne są zarówno na granicach ziaren ferrytu, jak i wewnątrz nich. Granice ziaren to obszary o mniejszej energii, gdzie często dochodzi do wydzielania się faz wtórnych oraz inicjacji pęknięć, na co zwracano uwagę przy analizie zdjęć mikroskopowych.

W celu określenia stanu degradacji i wytrzymałości analizowanych struktur przeprowadzono poszukiwanie defektów na podstawie uzyskanych obrazów. Nie wykryto żadnych oznak inicjacji uszkodzeń opisa-



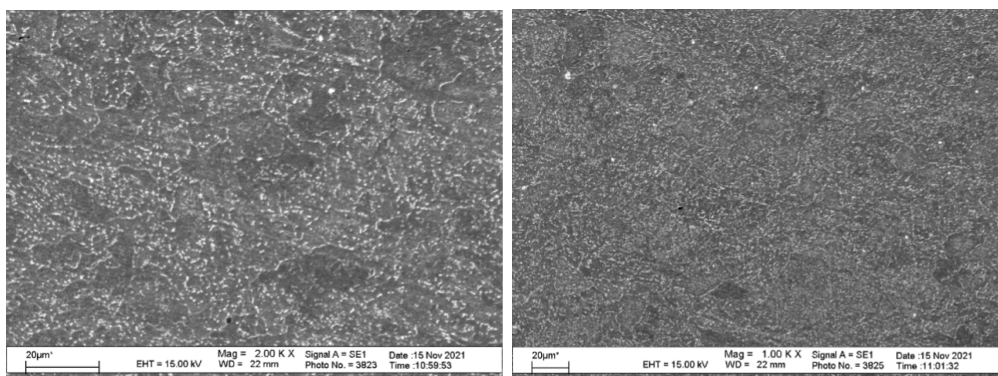
Rysunek 4.10: Zdjęcia SEM mikrostruktury próbki J1 - strona wewnętrzna



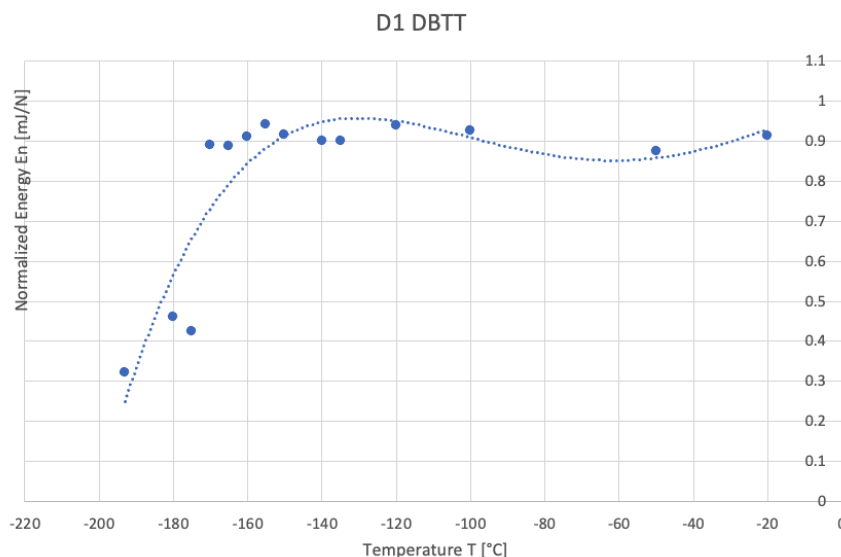
Rysunek 4.11: Zdjęcia SEM mikrostruktury próbki J1 - strona zewnętrzna

nych we wcześniejszym akapicie, które spełniałyby kryteria wykrywania defektów. W strukturze materiału nie stwierdzono nieciągłości ani mikropełnięć.

Poziom wyeksploatowania materiału (t/tr) został określony na 0.4 do 0.5, gdzie t oznacza czas dotychczasowej eksploatacji, a tr szacowany czas do zniszczenia i wskazuje, że materiał eksploatowany nie osiągnął jeszcze granicy swojej żywotności, co jest istotne w kontekście monitorowania stanu obiektów i konstrukcji takich jak zbiorniki i rurociągi.



Rysunek 4.12: Zdjęcia SEM mikrostruktury próbki J1 - przekrój poprzeczny



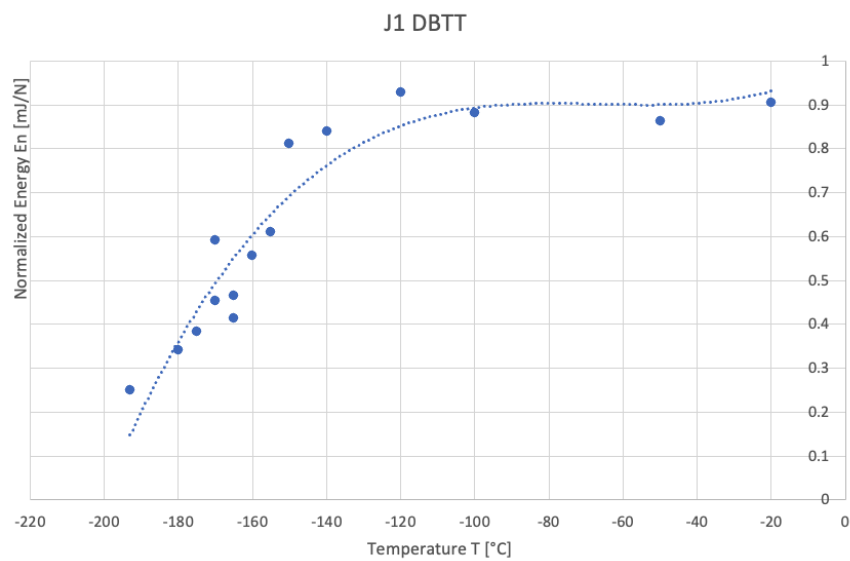
Rysunek 4.13: Krzywa DBTT dla materiału 10H2M w wariancie D1

DBTT i ocena wizualna Materiał 10H2M został poddany badaniom SPT w ekstremalnie niskich temperaturach, aby wyznaczyć granicę przejścia materiału w stan kruchy. Badania wykonano na zestawie próbek z materiału przepracowanego - J1 w ilości 15 sztuk i nowego - D1, gdzie przebadano 14 próbek. Do badania temperatury przejścia w stan kruchy wymagana jest, według normy [18] minimalna ilość próbek $N = 12$. W ramach analiz w laboratorium SPT w Centrum Energetyki AGH podjęto próby uzyskania wyników metodami alternatywnymi, nieopisywanymi w normie, przyjmując tezę, sugerującą minimalną liczbę próbek do badania jako $N = 2$.

Pierwszym wykonanym krokiem było przeprowadzenie testów SPT w niskich temperaturach, rozpoczynając od -20°C , a kończąc na około -200°C , z każdym kolejnym testem wykonanym w niższej temperaturze. Następnie zestawiono wyniki wyznaczonych wartości znormalizowanej energii E_n (wzór 2.25) na wykresie z temperaturą na osi X (wykres 4.13 oraz 4.14).

To podejście, z założenia standardowe, jest niezwykle czasochłonne ze względu na wymagane schładzanie próbki i komory testowej przy każdym pojedynczym teście. Dodatkowo, w przypadku niewielkiej ilości dostępnych próbek, wykonanie pełnego badania może okazać się niemożliwe. Próbki mogą zostać lepiej wykorzystane w badaniu standardowym SPT, dającym wiele więcej użytecznych wyników.

Aby poprzeć przyjętą wcześniej tezę, próbki poddano analizie wizualnej, przechodząc z wyznaczania granicy temperatury przejścia w stan kruchy na podstawie obliczeń. Dzięki analizie rodzaju pęknięcia próbki w danej temperaturze określono różnice pomiędzy typem propagacji pęknięcia próbki, jego wielkością oraz kształtem. To pozwoliło określić moment, na osi temperatury, w którym dany materiał przeszedł ze stanu plastycznego w kruchy. Wizualne rozróżnienie opisać można jako moment, w którym wgniecenie zadane przez stempel matrycy testowej wykonało wyraźnie mniejsze wgniecenie w próbce, ze względu na krótszy przebyty dystans wynikający ze zmniejszonego zakresu rozciągania materiału, tożsamego dla pęknięcia plastycznego. Pęknięcie przyjmuje prosty kształt, z mniejszą ilością ostrych ząbków niż tuż przed przejściem w stan kruchy. Otwarcie pęknięcia jest wyraźnie węższe po, a szeroko rozwarte tuż przed punktem przejścia. Na tej podstawie określono $T_{J1SP} = -160^{\circ}\text{C}$ (zdjęcie 4.15) oraz $T_{D1SP} = -170^{\circ}\text{C}$ (zdjęcie 4.16).



Rysunek 4.14: Krzywa DBTT dla materiału 10H2M w wariancie J1



Rysunek 4.15: Próbki J1 DBTT: -160C po lewej; -165C po prawej.



Rysunek 4.16: Próbki D1 DBTT: -175C po lewej; -170C po prawej.

Poza badaniami na materiale 10H2M, w ramach niniejszej pracy dalsze analizy temperatury przejścia w stan kruchy nie były kontynuowane w opisanym zakresie.

SPCT Jak opisano w poprzednim rozdziale, badanie SPCT wykonywano w laboratorium SPT w Centrum Energetyki AGH na zasadzie współpracy z Instytutem Metalurgii Żelaza w Gliwicach. Poprzez wspólne badania podjęto próbę skorelowania wyników SPT z klasycznym badaniem pełzania. Dla materiału 10H2M otrzymano następujące warunki z klasycznego badania pełzania:

- Temperatura badania: 700 C
- Naprężenie zadane próbce: 55 MPa
- Czas do zerwania próbki: 37 h

Na podstawie tych wartości przygotowano próby SPCT, przyjmując tę samą temperaturę testu, a obciążenie mierzone w tym przypadku w [N] wyznaczono z opisanego wcześniej przekształconego wzoru (3.1), gdzie dla 55 MPa, po przemnożeniu przez współczynnik Chakrabarty'ego $\Psi = 1,890 \text{ N/MPa}$, otrzymano wartość 104 N. Dla tych warunków przygotowano 8 próbek według standardu SPT. Próby przeprowadzono na maszynie testowej SPC 800 DLS. Wyniki prób przedstawiono w tabeli (Tablica 4.6). Z 8 pierwotnie przygotowanych próbek próbka numer 1 została wykorzystana jako próbka kalibracyjna, a próbki numer 4 i 5 sklasyfikowano jako wyniki odstające ze względu na całkowite przemieszczenie stempla znacznie mniejsze niż w przypadku pozostałych próbek, mimo zakończenia testu.

Material	10H2M			
Lp.	Time [h]	Creep rate [mm/h]	Displacement [mm]	Thickness [mm]
1	-	-	2.500	0.503
2	69	0.0032	2.500	0.503
3	33	0.0154	2.501	0.500
4	85	0.0005	1.269	0.504
5	4	0.0456	0.861	0.504
6	44	0.0103	2.502	0.504
7	50	0.0084	2.501	0.504
8	40	0.0062	2.504	0.504

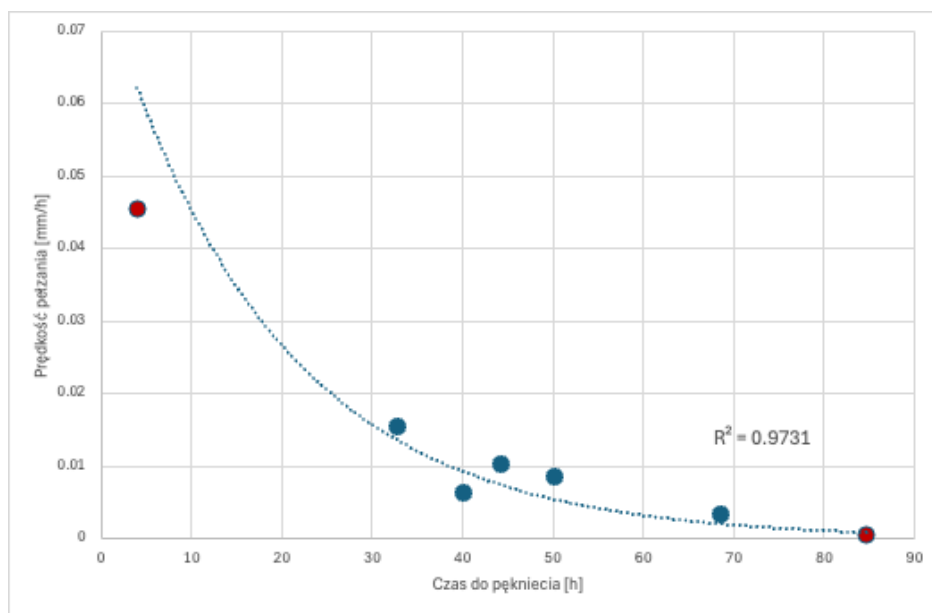
Tabela 4.6: Wyniki SPCT dla stali 10H2M

Uśredniony czas do pęknięcia w serii testów SPCT, nieuwzględniający dwóch wyników odstających, wyniósł 47 h, co jest wartością o 10 h większą niż zaraportowany przez IMŻ czas do pęknięcia. Biorąc pod uwagę skalę czasów do pęknięcia w ogólnych testach pełzania, jest to odchylenie stosunkowo niewielkie, jednak wskazuje na konieczność doprecyzowania dalszych korelacji wyników, eksploatując inne wartości znane z literatury, jak np. równanie 2.11 przedstawione w pracy [95], przedstawiające stosunek siły SPCT do naprężenia UCT jako $\alpha = 2.1$ dla atmosfery argonu, która to jest wykorzystywana w trakcie badania

na maszynie SPC 800 DLS, co efektywnie podniosłoby obciążenie w teście SPCT, potencjalnie skutkując krótszym czasem do pęknięcia.

Na podstawie wyników przedstawionych w tabeli (4.6) przygotowano wykres (4.17) zależności zarejestrowanej najmniejszej prędkości pełzania do czasu trwania testu. Z wykresu wynika, że wykładnicza linia trendu opisana równaniem (4.1) daje zgodność rzędu 97% dla przeanalizowanych próbek.

$$f(x) = 0.0768e^{-0.053x} \quad (4.1)$$



Rysunek 4.17: Wykres zależności minimalnej prędkości pełzaniu od czasu do pęknięcia próbki w teście SPCT

Ze względu na nakierowanie tematyki pracy na inne konkretne cele, nie zdecydowano się zgłębiać dalej badań nad zagadnieniem SPTC.

4.1.2 Stal 13HMF

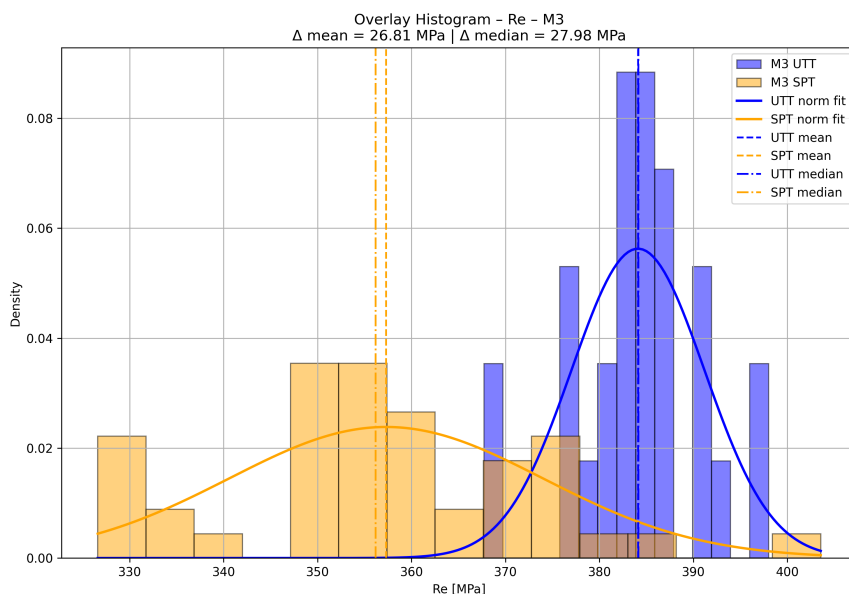
Podobnie jak w przypadku pierwszego analizowanego materiału, to samo podejście statystyczne zastosowano dla próbek ze stali 13HMF. Materiał w stanie wyjściowym, nieprzepracowanym, oznaczono M3. Materiał w stanie silnie przepracowanym w warunkach użytkowych, tożsamych z typem stali kotłowej, który po 200 tysiącach godzin został wyłączony z użytku, oznaczono R1. Próbki do badania SPT przygotowano w pełnej zgodności z europejskim standardem EN 10371:2021, zachowując wymaganą tolerancję grubości próbki z dokładnością do 0.005 mm. Wstępne uśrednione wyniki z obu metod badawczych przedstawiono w tabeli 4.7. Tabela zawiera, jak w przypadku poprzedniego materiału, wartości statystyczne dla analizowanych grup materiałów i ich właściwości, a jej treść podsumowuje pierwszy krok analizy statystycznej.

Wyniki rozkładów w formie zbiorczej dla obydwu grup materiału: R1 i M3 oraz dla obydwu parametrów: R_e i R_m przedstawiono na histogramach 4.18 do 4.21. Całkowita populacja próbek materiału 13HMF dobrze wpisuje się w granice rozkładu normalnego, co potwierdza porównywalną wiarygodność wyników

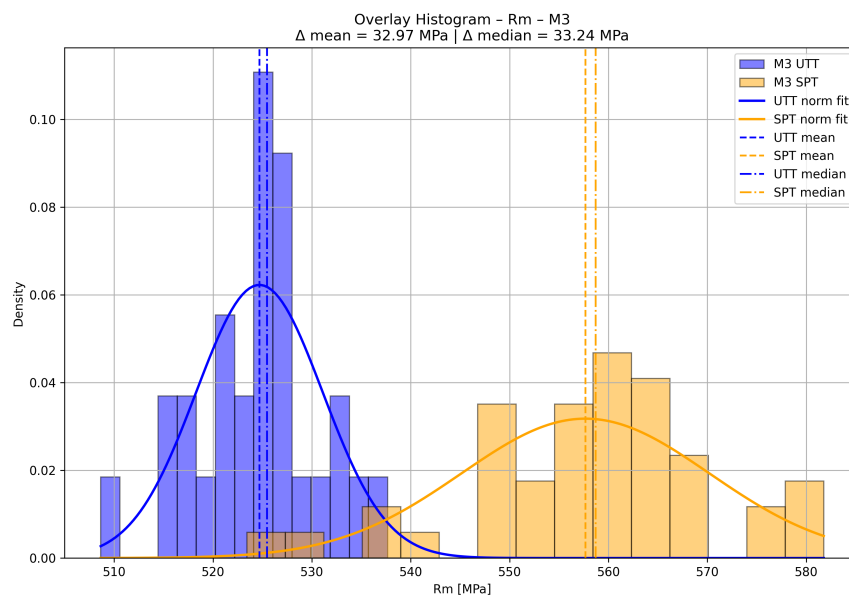
Property	Material	Mean [MPa]	[%] AVG Δ to UTT	Std Dev	AD Statistic	Shapiro p	Normality	Best-Fit Dist	N	Outliers Removed
Re	M3 SPT	357	-7.0%	16.7	0.5694	0.1901	Normal	norm	44	2
Re	M3 UTT	384	-	7.1	0.4931	0.2487	Normal	norm	28	2
Re	R1 SPT	437	-2.5%	28.3	0.7690	0.0479	Not Normal	weibull_mi	32	0
Re	R1 UTT	448	-	9.7	0.0848	0.9996	Normal	norm	28	1
Rm	M3 SPT	558	6.3%	12.6	0.5941	0.1428	Normal	norm	44	2
Rm	M3 UTT	525	-	6.4	0.2379	0.9163	Normal	norm	28	2
Rm	R1 SPT	650	1.9%	19.4	0.4982	0.0885	Normal	norm	32	0
Rm	R1 UTT	638	-	10.9	0.3323	0.7139	Normal	norm	29	0

Tabela 4.7: Średnie wartości wyników badań metodami SPT i UTT dla materiału 13HMF.

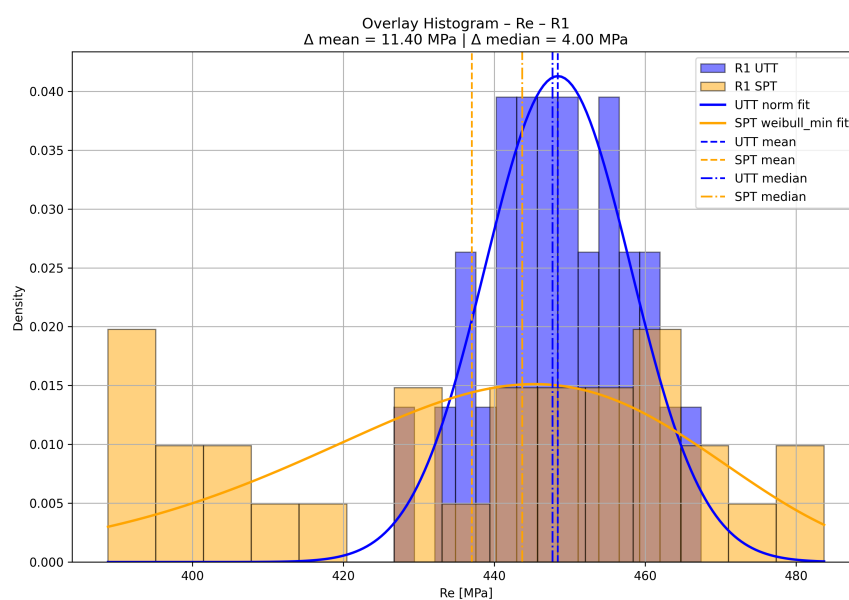
SPT względem wyników uzyskiwanych metodą jednoosiowego rozciągania. Wyjątkiem jest grupa R1 w badaniu SPT dla parametru Re, gdzie mimo zgodności statystycznej z rozkładem normalnym, lepszym dopasowaniem dla tego zbioru mogłoby być wykorzystanie rozkładu *Weibull Minimum Extreme Value* ze względu na nagromadzenie wyników przy krańcu rozkładu. Zauważalną różnicę stwierdzono w odchyleniu standardowym wartości wytrzymałości na rozciąganie (Ultimate Strength), gdzie dla metody SPT odnotowano większy rozrzut oraz zakres uzyskanych wartości niż w przypadku UTT. W przypadku granicy plastyczności (Yield Strength) rozrzut wartości dla metody SPT okazał się jeszcze większy, co wynika z charakterystyki tego parametru. Dla badanego materiału zauważono bardzo dobrą i stabilną korelację wyników pomiędzy obiema metodami, SPT i UTT. Charakteryzuje się to przesunięciem wartości średnich wyników SPT badania granicy plastyczności o około 15 MPa w górę względem wyników UTT. Korelacja ta jest dalej umocniona w przypadku analizy granicy wytrzymałości materiału, która z zasady jest stabilniejszym parametrem, a jej określanie nie pozostawia wątpliwości co do dokładności metody. W przypadku materiału nieprzepracowanego, metoda SPT wykazała o około 35 MPa wyższe średnie wyniki granicy wytrzymałości niż metoda UTT. Ten trend przesunięcia zmniejszył się jednak do 12 MPa dla materiału przepracowanego.



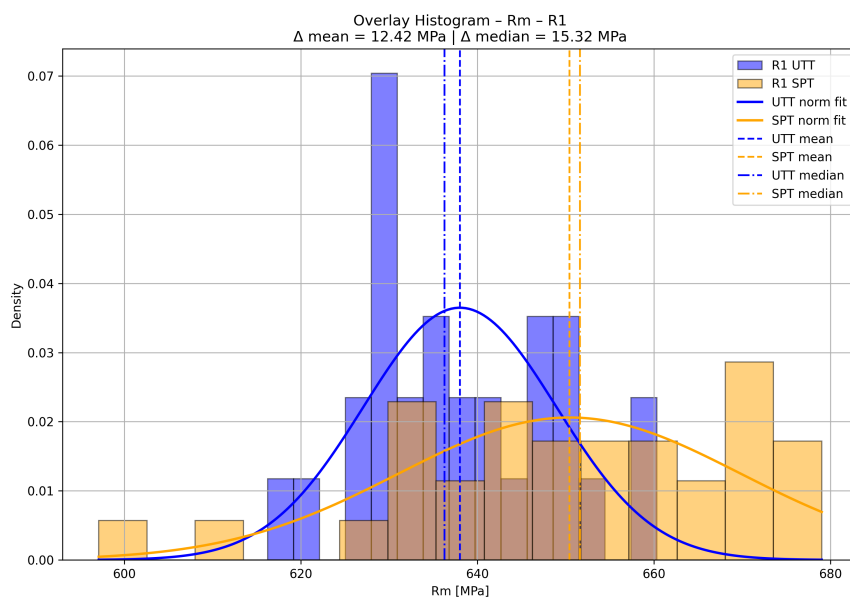
Rysunek 4.18: Histogramy wyników M3 - Granica plastyczności



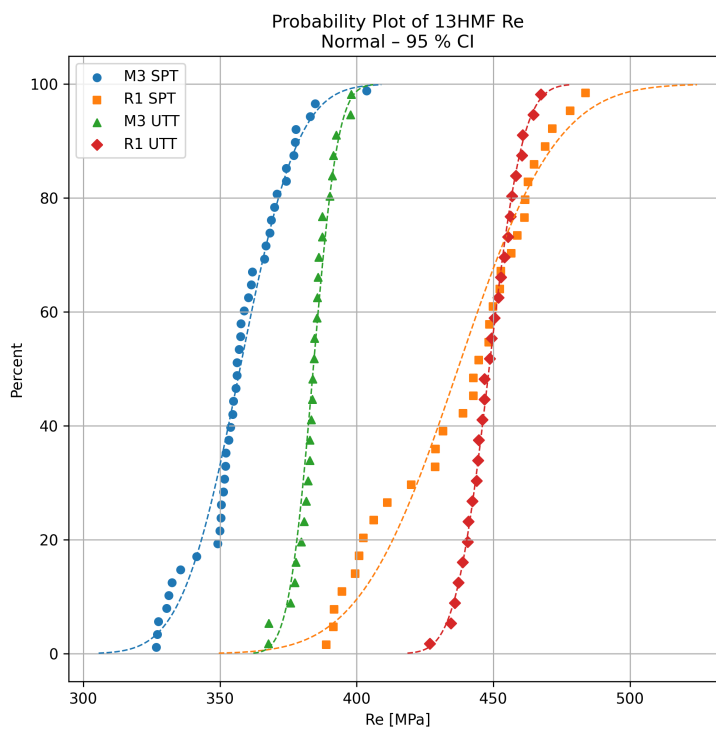
Rysunek 4.19: Histogramy wyników M3 - Granica wytrzymałości



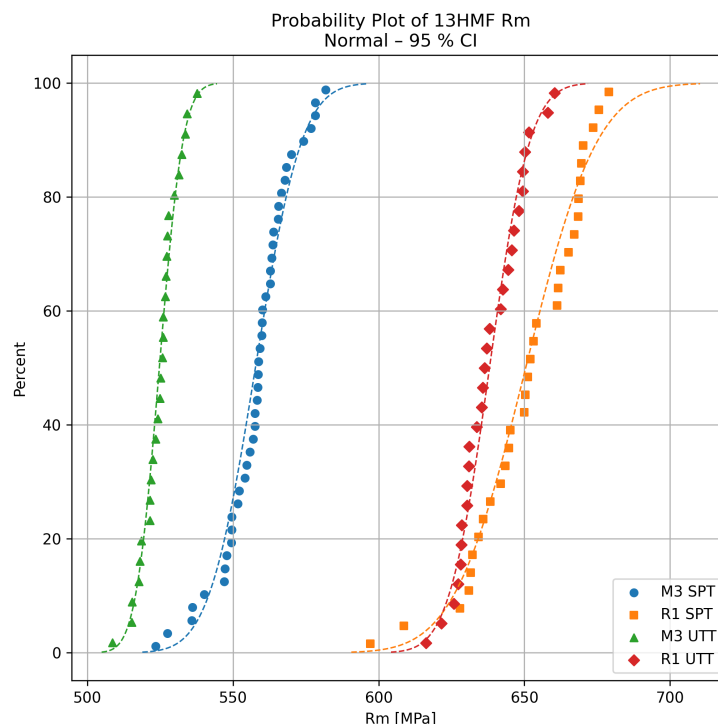
Rysunek 4.20: Histogramy wyników R1 - Granica plastyczności



Rysunek 4.21: Histogramy wyników R1 - Granica wytrzymałości



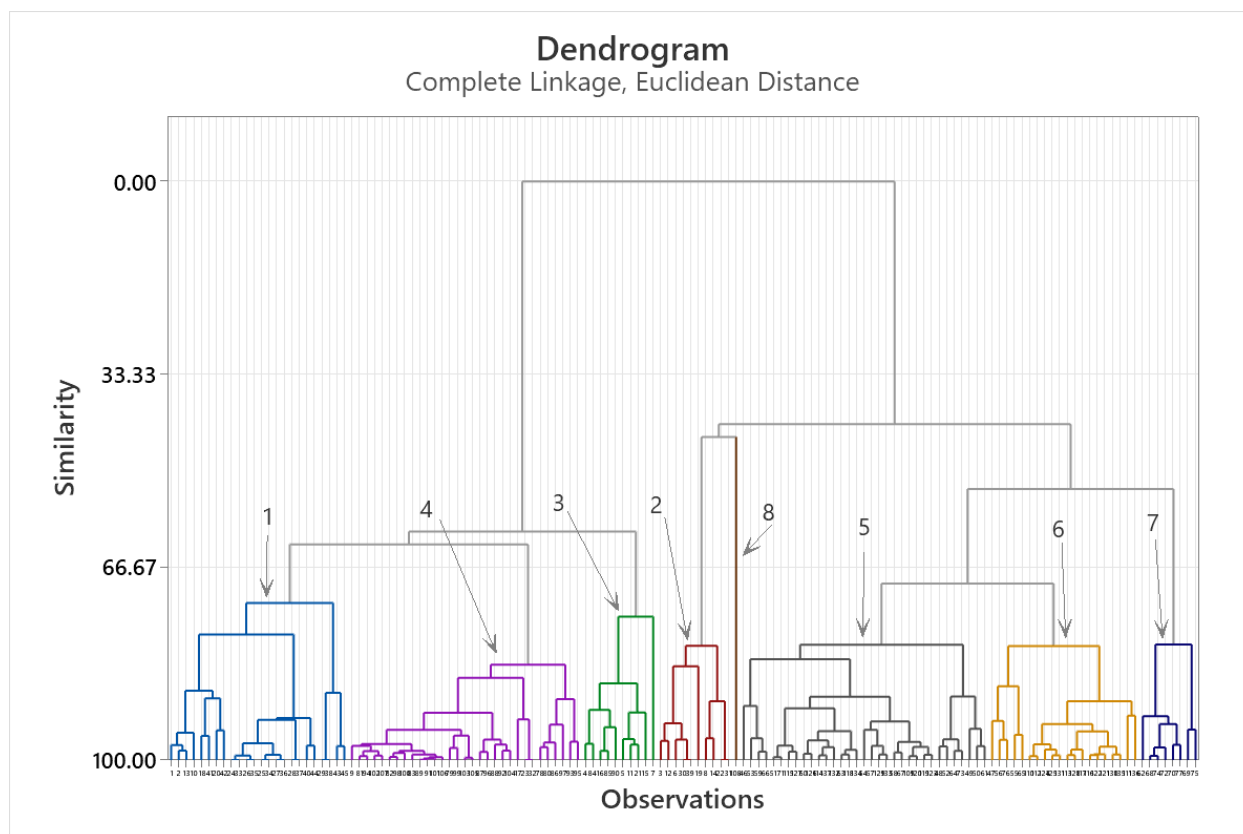
Rysunek 4.22: Wykresy prawdopodobieństwa rozkładu dla stali 13HMF - Granica plastyczności



Rysunek 4.23: Wykresy prawdopodobieństwa rozkładu dla stali 13HMF - Granica wytrzymałości

Ogólna analiza materiału 13HMF wykazała, że po 200 000 godzinach eksploatacji nastąpiło wyraźne umocnienie struktury materiału, które prawidłowo odzwierciedlają obie metody badawcze. Otrzymane wartości wskazują jednak, że próba SPT jest znacznie bardziej czuła na zmiany mikrostrukturalne oraz mikroodkształcenia, co czyni ją doskonałym narzędziem do wykrywania subtelnych różnic wynikających z długotrwałego użytkowania. Na wykresach prawdopodobieństwa widoczna jest stała rozbieżność wartości R_e pomiędzy metodami SPT i UTT dla grupy M3, tj. materiału nieprzepracowanego, która dalej zanika dla materiału przepracowanego i widoczne jest duże pokrycie w centralnym segmencie rozkładu, czego efektem jest niewielka różnica wartości mediany dla obu grup (Rysunek 4.22). Trend rozbieżności potwierdza się również dla parametru R_m pomiędzy próbkami z materiału nowego, a zmniejsza się w przypadku grupy R1, tj. materiału przepracowanego (Rysunek 4.23)

W kolejnym etapie przeprowadzono analizę klastrową na całej populacji próbek. Jako podstawę obserwacji wyznaczono 70% stopień podobieństwa jako kryterium akceptacji do korelacji obu metod testowych, SPT i UTT, i ich wyników. Wyróżniono osiem klastrow dla materiału 13HMF (Tablica 4.8). Podział wyróżnia klastry 1, 4, 5 i 6, które reprezentują odpowiednio grupy M3 SPT, M3 UTT, R1 SPT oraz R1 UTT. Do klasteryzacji przyjęto wszystkie 137 próbek poddanych analizie statystycznej przed oddzieleniem próbek skrajnych. Co warto podkreślić, do klasteryzacji próbki podchodzą z parami parametrów, gdzie ich charakterystyka rozważana jest na podstawie wartości R_e oraz R_m . Na podstawie dendrogramu oraz klastrow wynikowych widoczne są liczne mniejsze grupy do 10 próbek na klastrow, które pochodzą potencjalnie ze skrajnych ogonów widocznych w zbiorach danych podczas wcześniejszych etapów analizy statystycznej, jak przy budowie histogramów. Opisuje to efekt wspomnianego wcześniej braku silnego kontrastu pomiędzy metodami, kiedy rozdzielone zostaną wartości skrajne. Pozwala to jednocześnie odkryć rzeczywiste różnice



Rysunek 4.24: Dendrogram z analizy klastrow dla materiału 13HMF

w wartościach uzyskanych obiema metodami i podkreślić wcześniejszą obserwację z materiału 10H2M, mówiącą o wynikach SPT będących zwykle wyższymi niż UTT.

Variable	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7	Cluster8	Grand Cent.
Re	342.075	436.130	357.620	386.052	456.822	437.035	396.938	527.200	406.083
Rm	549.204	562.660	506.585	527.880	655.641	634.377	625.900	628.090	585.377
N	24	10	10	31	33	20	8	1	137

Tabela 4.8: Centroidy klastrow materiału 13HMF.

Wygenerowany przy klasteryzacji dendrogram (Rysunek 4.24) przedstawia wyraźny podział pomiędzy oba stany materiału, nowy i w stanie przepracowanym. Dendrogram pozwala wizualnie zinterpretować podobieństwa na różnych poziomach, pokazując, że podział na materiały M3 i R1 następuje natychmiastowo, a dopiero dalej przy granicy około 65% rozróżniamy podział dla materiału M3 na rodzaj testu, z którego pochodzą próbki. Podobna sytuacja ma miejsce na poziomie około 70% dla materiału R1. Po przekroczeniu podobieństwa 80% gwałtownie wzrasta gęstość formacji klastrow, tworząc mniejsze, nierozróżnialne podgrupy w obszarach określonych już klastrow pochodzących z przyjętego podziału na stan i rodzaj testu danego materiału. Podkreśla to dobre podobieństwo obu metod testowych.

4.1.3 Stal 15HM

Do analizy trzeciego badanego w pracy materiału, jakim jest stal 15HM, wybrano dwie grupy H1 oraz T1, gdzie T1 odpowiada materiałowi w stanie nowym, nieprzepracowanemu, a T1 materiałowi po 130 tysiącach godzin pracy. Podobnie jak w przypadku stali 13HMF do badań SPT wykorzystano jedynie próbki w oryginalnym zakresie tolerancji grubości. Wyniki z analizy statystycznej przedstawiono w tabeli 4.9. Już na początku warto zwrócić uwagę, że metoda klasyfikacji wyników odstających zdeterminowała aż 5 takich wyników dla parametru Re w grupie T1 UTT. Wskazuje to na silny wpływ samego materiału na wyniki podczas badania metodą o ugruntowanej pozycji na rynku.

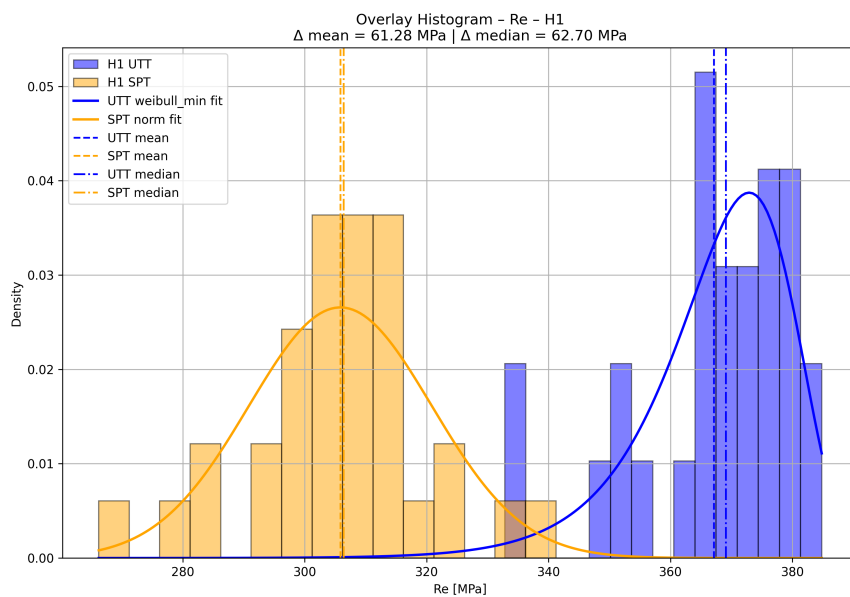
Property	Material	Mean [MPa]	[%] AVG Δ to UTT	Std Dev	AD Statistic	Shapiro p	Normality	Best-Fit Dist	N	Outliers Removed
Re	H1 SPT	306	-16.6%	15	0.4509	0.5472	Normal	norm	33	2
Re	H1 UTT	367	-	13.5	1.0143	0.0026	Not Normal	weibull_mi	28	2
Re	T1 SPT	410	-1.9%	46.4	0.3044	0.5684	Normal	norm	33	0
Re	T1 UTT	418	-	21.2	1.6097	0.001	Not Normal	lognorm	25	5
Rm	H1 SPT	517	5.7%	8.8	0.8299	0.012	Not Normal	beta	31	2
Rm	H1 UTT	489	-	10	0.5645	0.1127	Normal	norm	28	2
Rm	T1 SPT	593	9.6%	18.5	0.3326	0.5556	Normal	norm	31	0
Rm	T1 UTT	541	-	15	0.8116	0.0508	Normal	norm	30	0

Tabela 4.9: Średnie wartości wyników badań metodami SPT i UTT dla materiału 15HM.

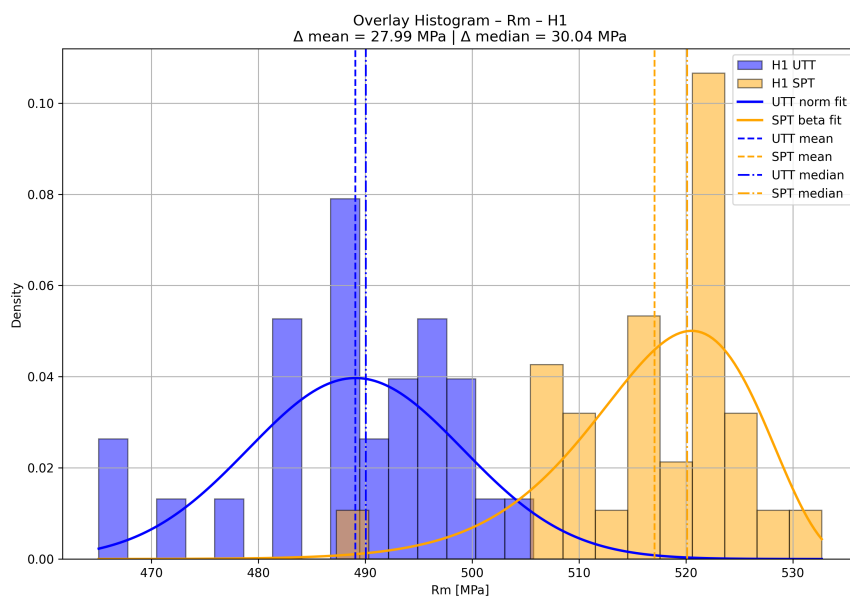
Przechodząc z opisowej do graficznej analizy, skupiono się na histogramach (Rysunki 4.25, 4.26, 4.27, 4.28 zestawiających dane ze obydwu grup materiału i metod pomiarowych. Dla materiału H1, będącego materiałem nieeksploatowanym, duże przesunięcie wartości odczytanej z badania SPT i UTT dla parametrów Re. Rozbieżność wyników średnich wartości sięga ponad 60 MPa, mimo że pojedyncze wyniki wykazują wysoką przynależność do swoich grup, sądząc po widocznym rozkładzie zbliżonym do normalnego. Wyjątkiem od tej zasady jest porównanie metod SPT i UTT dla wartości Re materiału T1 na histogramie 4.27, gdzie mimo szerszego rozkładu wyników SPT wartość średnia różni się jedynie o około 8 MPa względem badania UTT. W przypadku porównania mediany ta wartość sięga już niemal 19 MPa, jednak należy tu zwrócić uwagę, że dla UTT analizowano tylko 25 wartości, ponieważ 5 zostało sklasyfikowanych jako wartości skrajnie odstające.

Do dalszego porównania wizualnego zarówno metod badawczych, jak i klas materiału, z podziałem pomiędzy wartości granicy plastyczności (Rysunek 4.29), a granicy wytrzymałości (Rysunek 4.30), wskazują ponownie, że metoda SPT wykazuje większą czułość na zmiany w wartościach granicy plastyczności, uwidocznioną poprzez wyższą niż w przypadku metody UTT zmianę wartości pomiędzy materiałem nowym a przepracowanym. Dla granicy wytrzymałości obserwowana zmiana wartości pomiędzy stanem materiału ponownie jest wyższa. W tym przypadku jest jednak obrazowana jako ogólne wykazywanie wyższych wyników parametru Rm względem metody UTT, co potwierdza wcześniej obserwowane zależności dla badanych materiałów o stałych, wyższych odczytach granicy wytrzymałości.

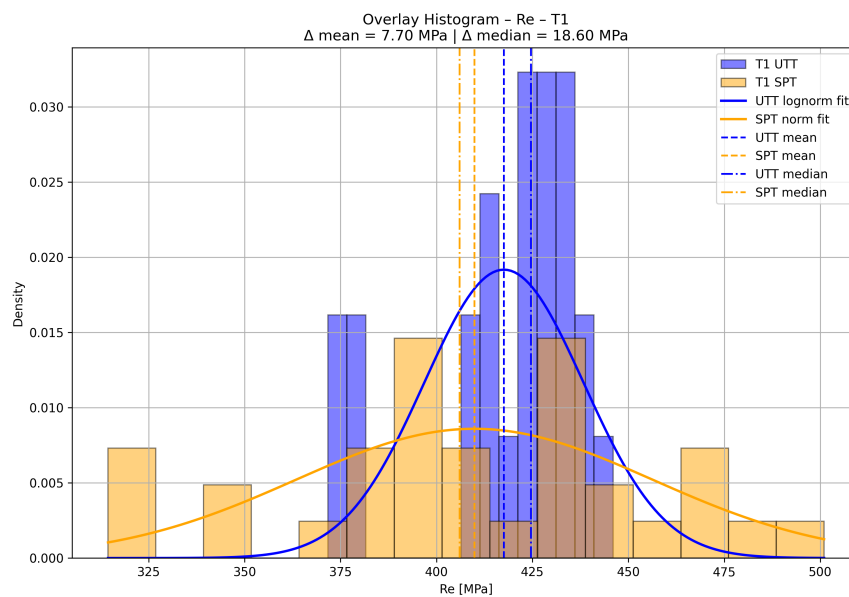
W analizie klastrow z wykorzystaniem narzędzia generowania dendrogramów, podobnie jak w przypadku dwóch poprzednich opisywanych materiałów, przyjęto jako dane wejściowe całe populacje danych bez wstępnego oddzielania wartości skrajnych, pozwalając, aby podział nastąpił dopiero na etapie formowania klastrow. Łącznie badaniu poddano 127 próbek z materiału 15HM. Jako kryterium podobieństwa



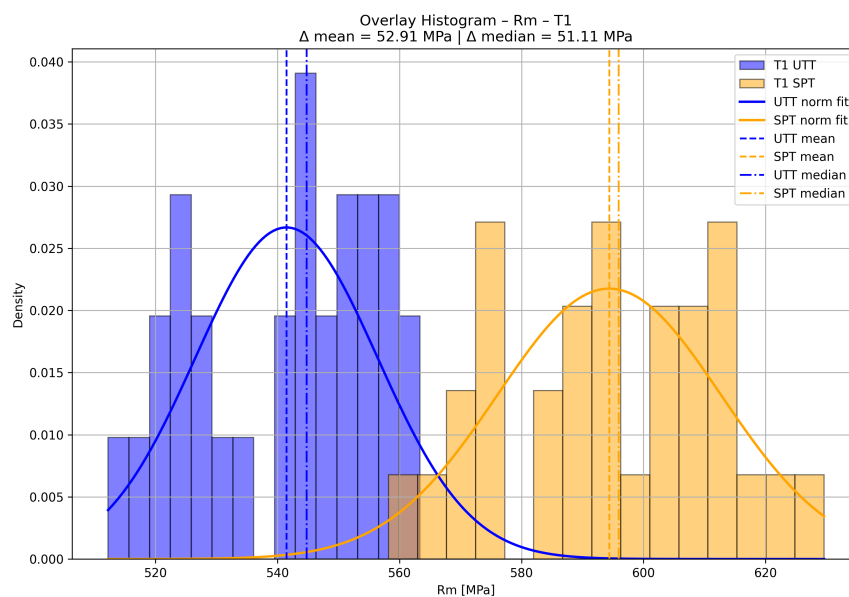
Rysunek 4.25: Histogramy wyników H1 - Granica plastyczności



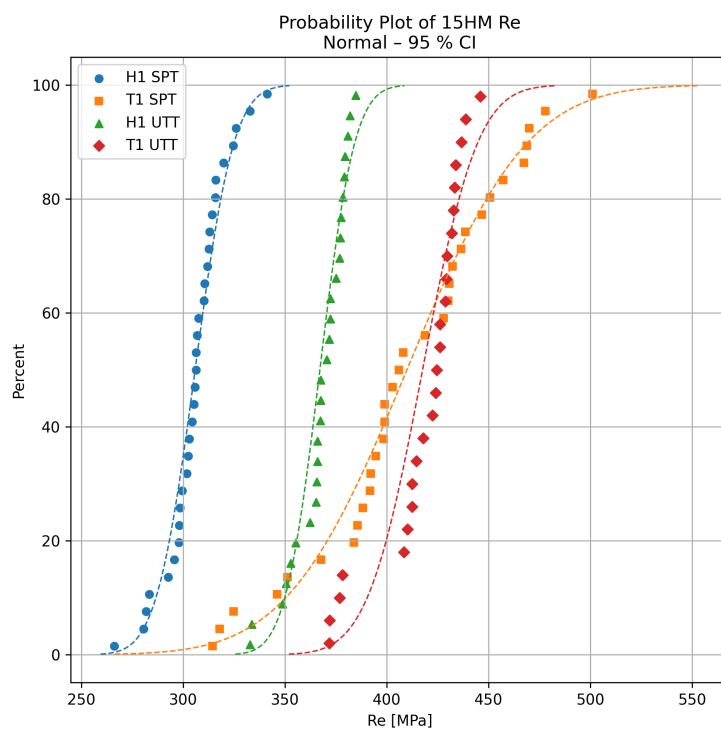
Rysunek 4.26: Histogramy wyników H1 - Granica wytrzymałości



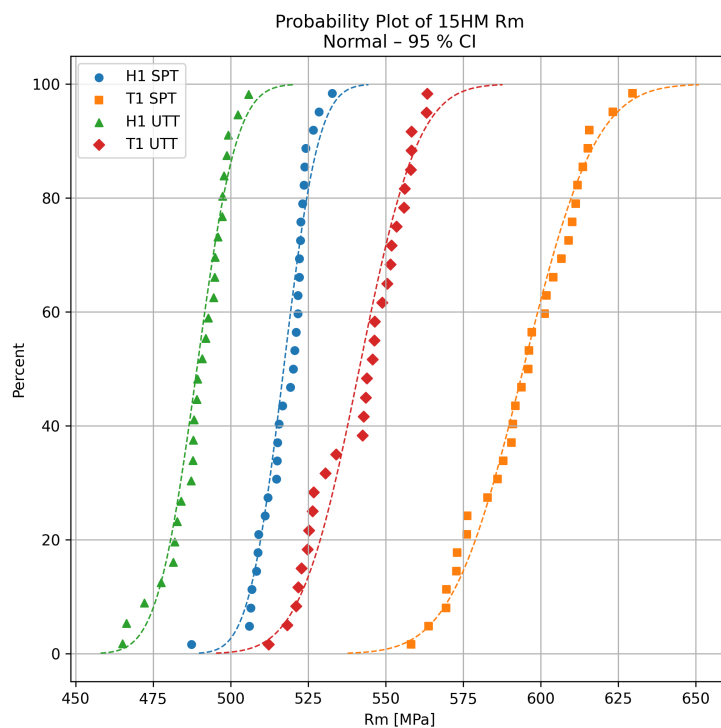
Rysunek 4.27: Histogramy wyników T1 - Granica plastyczności



Rysunek 4.28: Histogramy wyników T1 - Granica wytrzymałości



Rysunek 4.29: Wykresy prawdopodobieństwa rozkładu dla stali 15HM - Granica plastyczności



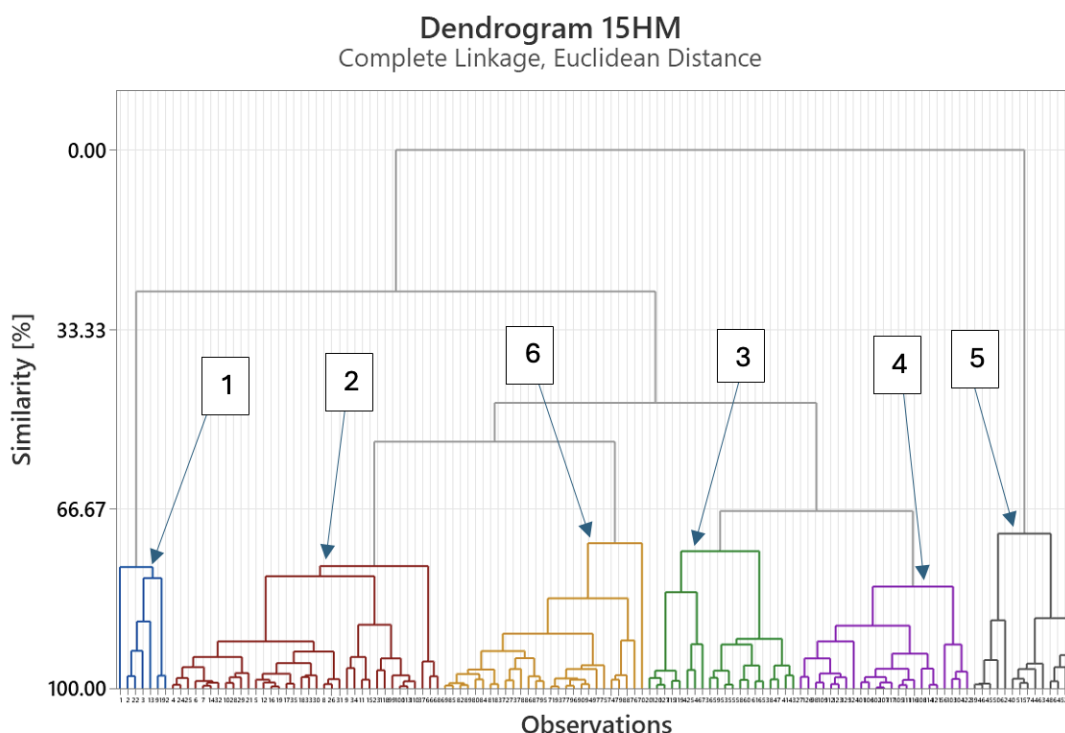
Rysunek 4.30: Wykresy prawdopodobieństwa rozkładu dla stali 15HM - Granica wytrzymałości

przyjęto wartość 70% podobieństwa, a w wyniku analizy otrzymano 6 osobnych klastrow (Tabela 4.10). Tak więc, zwracając uwagę na centroidy każdego z klastrow, klaster numer 2 odpowiadał wartościom grupy H1 SPT, podczas gdy numery 3 oraz 5 stanowiły w większości podział skrajnych wartości grupy T1 SPT ze względu na najwyższe wśród analizowanych grup materiałów wartości odchylenia standardowego, świadczące o szerokim zakresie wartości wyników pomiarów. Klaster numer 4 odpowiadał grupie T1 UTT, a numer 6 grupie H1 UTT. Klaster 1, zawierający w sobie 7 próbek, zawierał wartości skrajne odstające od przyjętych podziałów. Podział zaprezentowany graficznie na dendrogramie (Rysunek 4.31) wskazuje, mię-

Variable	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Grand centroid
Re	297.643	316.372	384.16	425.717	451.257	371.885	372.489
Rm	460.669	521.561	568.72	550.280	610.686	488.879	533.709
N	7	36	20	23	14	27	127

Tabela 4.10: Centroidy klastrow materiału 15HM.

dzy innymi, że podział pomiędzy wynikami z testu SPT a UTT dla materiału H1 nastąpił już na poziomie około 55%. W przypadku materiału T1 podział ten nastąpił później, bo aż na poziomie 67%, jednak dotyczył tylko części próbek SPT należących do klastra 3, ponieważ klaster 5 wydzielił się już na samym początku analizy.



Rysunek 4.31: Dendrogram z analizy klastrow dla materiału 15HM

4.2 Modelowanie danych z użyciem Sieci Neuronowej i analiza wyników predykcji

Proces predykcji danych UTT z danych SPT za pomocą modelowania z wykorzystaniem Sieci Neuronowych wykonano dla obu przygotowanych modeli sieci, modelu w architekturze CNN oraz w architekturze MLP. Wartości pozyskane w obu metodach zestawiono podobnie jak w poprzedniej sekcji w przypadku SPT, z badaniem UTT. Wykresy porównawcze oraz dane zestawione w tabelach przedstawiono dla każdego z badanych materiałów. Należy dodać, iż wyniki predykcji z sieci neuronowych uwzględniają współczynnik kompensacji związany z grubością próbki z testu SPT, na podstawie którego dokonano predykcji, w celu wprowadzenia naturalnych zakłóceń wynikających z zakresu tolerancji próbek SPT. Zaklasyfikowane one były jako zgodne z zaproponowaną tolerancją poszerzoną względem normy do $\pm 0.01mm$ oraz o wysokiej jakości powierzchni. Wszystkie wykresy przygotowane przez sieci neuronowe były poddawane automatycznej analizie w celu wyekstrahowania wyników z wygenerowanych surowych danych, przekształconych na wykresy. Metoda generowania i odczytywania parametrów tu analizowanych była jednakowa dla obu sieci neuronowych. Wszystkie wyniki zostały poddane walidacji na specjalnie odseparowanych próbkach niebiorących udziału w uczeniu sieci neuronowej. To sprawdzenie jest kluczowym narzędziem oceny wiarygodności i stabilności uzyskanych wyników. Wyniki przedstawione oceniają zdolności sieci do generalizacji na nieznanych zbiorach danych.

4.2.1 Stal 10H2M

Dla materiału 10H2M jako dane wejściowe do sieci neuronowych wykorzystano 47 próbek materiału z D1 oraz 46 próbek materiału J1. Podsumowane wyniki analizy dla materiału 10H2M przedstawiono w tabeli 4.11. Tabela podaje wartości średnie dla poszczególnych materiałów i parametrów. Dodatkowa kolumna wskazuje procentową odchyłkę względem wartości otrzymanych w UTT. Zauważyć można, że odchyłki dla granicy wytrzymałości pomiędzy źródłami wyników są marginalne dla obu wariantów sieci. Największe różnice obserwuje się dla granicy plastyczności materiału D1, i jest to ten sam kierunek odchyłki dla obu typów sieci neuronowych. Natomiast dla materiału J1 sieci CNN występuje trend przeciwny, aczkolwiek o mniejszej amplitudzie. Sieć MLP z powodzeniem i doskonałą dokładnością poradziła sobie z predykcją granicy plastyczności materiału J1. Należy tu jednak nadmienić, iż wariancje dla grupy D1 wynikają z faktu, że dla materiału 10H2M jako całości krzywe materiałowe z testu UTT są niezwykle podobne, a test SPT wychwytywał różnice w zachowaniu materiału, których wyniki UTT nie ujawniały. Co więcej, należy zwrócić uwagę nie tylko na sam odczytany wynik predykcji granicy plastyczności materiału D1 lub J1, ale również na otrzymane krzywe materiałowe.

Wyniki zestawienia próbek ze zbiorów testowych przedstawiono w tabeli (4.12). Wyniki tam zestawione intencjonalnie nie są objęte współczynnikiem korekcji grubości próbki SPT i są to wyniki surowe, co jest zasadą dla wszystkich trzech materiałów w tym rozdziale. Dla obu grup materiału było po 10 próbek CNN i 5 próbek MLP. Ponownie widoczne jest zachowanie odwzorowywane przez całą populację, tj. mieszanie zachowań D1 i J1 przez wartości odczytane z wykresów. Obrazuje to, jak delikatne różnice występują pomiędzy tymi dwoma materiałami.

Property	Material	Mean [MPa]	Median [MPa]	Std Dev	[%] AVG Δ to UTT	N	Outliers Removed	Usable N [%]
Re	D1 CNN	330	328	17.23	-7.8%	47	0	100%
Re	D1 MLP	318	315	12.09	-11.2%	47	0	100%
Re	D1 UTT	358	348	32.02	-	31	0	100%
Re	J1 CNN	337	341	16.65	2.1%	43	3	93%
Re	J1 MLP	328	329	10.43	-0.6%	46	0	100%
Re	J1 UTT	330	331	9.57	-	29	1	97%
Rm	D1 CNN	529	529	4.45	0.6%	47	0	100%
Rm	D1 MLP	522	522	4.38	-0.8%	46	1	98%
Rm	D1 UTT	526	523	11.57	-	31	0	100%
Rm	J1 CNN	523	522	3.41	-0.6%	41	5	89%
Rm	J1 MLP	517	517	3.51	-1.7%	41	5	89%
Rm	J1 UTT	526	525	4.52	-	30	0	100%

Tabela 4.11: Tabela porównawcza dla wyników predykcji dla materiału 10H2M

Material	D1				J1			
Property	Re		Rm		Re		Rm	
Unit	[MPa]	Δ [%]	[MPa]	Δ [%]	[MPa]	Δ [%]	[MPa]	Δ [%]
UTT	358	-	526	-	330	-	526	-
CNN	336	-6.1%	529	0.5%	319	-3.4%	518	-1.6%
MLP	318	-11.2%	529	0.5%	327	-0.9%	517	-1.7%

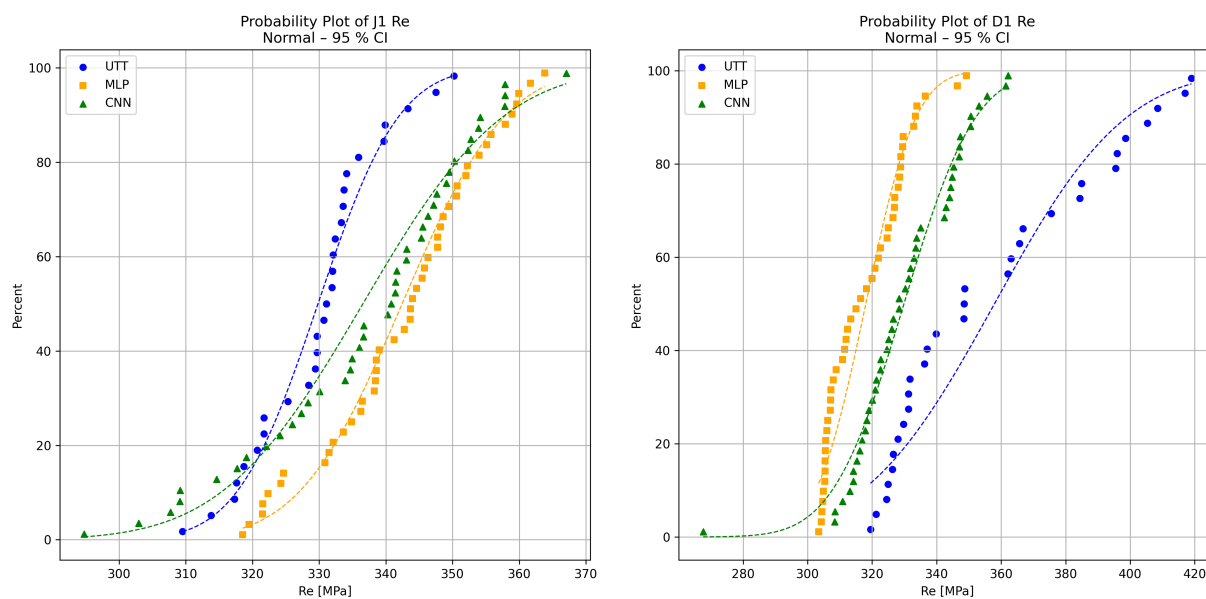
Tabela 4.12: Wyniki walidacji próbek ze zbioru testowego 10H2M

Te same dane z sieci neuronowych zostały dalej ocenione nie tylko pod względem zgodności z UTT, ale także pod względem statystycznego rozkładu populacji, co ujęto w tabeli 4.13. Zauważyć można, że dla materiału 10H2M nie istnieje zasada wskazująca, że którakolwiek z metod pozyskania wyników charakteryzuje się znaczącymi odchyłkami względem rozkładu normalnego. Jedynie w przypadku parametru Re dla grupy J1 populacja nie jest skrzywiona, ze względu na charakterystykę tej grupy materiału, gdzie wszystkie wzorcowe wykresy UTT wykazywały gładki przebieg, co przełożyło się na dobre odwzorowania za pomocą sieci neuronowych.

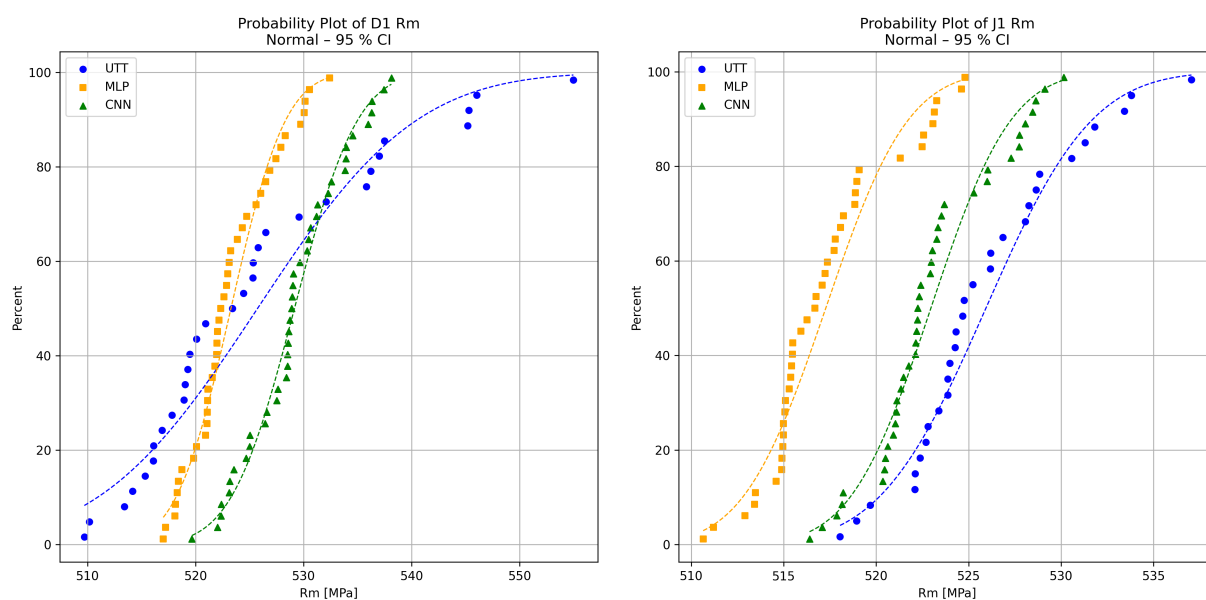
Property	Material	AD Statistic	Shapiro p	Normality	Best-Fit Dist	N
Re	D1 CNN	0.602	0.015	Not Normal	lognorm	47
Re	D1 MLP	1.496	0.001	Not Normal	weibull_min	47
Re	D1 UTT	1.18	0.005	Not Normal	weibull_min	31
Re	J1 CNN	0.644	0.13	Normal	norm	43
Re	J1 MLP	0.311	0.407	Normal	norm	46
Re	J1 UTT	0.443	0.631	Normal	norm	29
Rm	D1 CNN	0.432	0.39	Normal	norm	47
Rm	D1 MLP	0.499	0.164	Normal	norm	46
Rm	D1 UTT	0.876	0.029	Not Normal	weibull_min	31
Rm	J1 CNN	0.912	0.072	Normal	norm	41
Rm	J1 MLP	1.123	0.022	Not Normal	lognorm	41
Rm	J1 UTT	0.475	0.366	Normal	norm	30

Tabela 4.13: Podsumowanie testów dopasowania normalnego dla stali 10H2M

Rozpatrując dalej wyniki odczytów granicy plastyczności w postaci wykresów prawdopodobieństwa (Rysunek 4.32) zauważyć można, że dla materiału D1 obie metody predykcyjne przedstawiają bardzo ściśle zgrupowane wyniki, w kontraście do widocznego rozkładu tego parametru dla oryginalnych wyników UTT, które rozrzucone są w małych klastrach na szerokości 100 MPa. Co istotne, jak wspomniano wcześniej, wyniki z dolnego zakresu Re dla D1 UTT łudząco przypominają wyniki zestawione tuż obok dla J1 wycentrowane w okolicach 330 MPa. Dla samego materiału J1 sieć CNN w dolnym zakresie wskazuje dokładne przewidziane wartości Re, jednak w całym spektrum rozkład danych jest szerszy niż dla MLP, która wykazała dokładną predykcję rozkładu występującego w wzorcowej populacji UTT. Granica wytrzymałości dla materiału D1 UTT, podobnie jak granica plastyczności, jest bardziej zwarta w dolnym zakresie, a dopiero poszczególne próbki w małych klastrach sięgają wyższych wartości. Obie sieci dopasowały się do wartości uśrednionych, jednak z różnym efektem. Zważywszy na wielkość różnicy w rozkładzie wartości średnich, przyjmuje się to za bardzo dobre dopasowanie z odchyleniem poniżej 1% wartości. Dla grupy J1 widoczne jest niemal idealne odwzorowanie schematu rozkładu próbek UTT przez próbki CNN z różnicą wartości średnich poniżej 1%. Mimo iż w pracy dołożono wszelkich starań, aby metody odczytywania automatycznego wartości granicy z wygenerowanej krzywej były jak najlepsze, dopiero wiedza ekspercka i indywidualne podejście do każdego z wykresów naprężenie-odkształcenie pozwolą dokonać właściwej

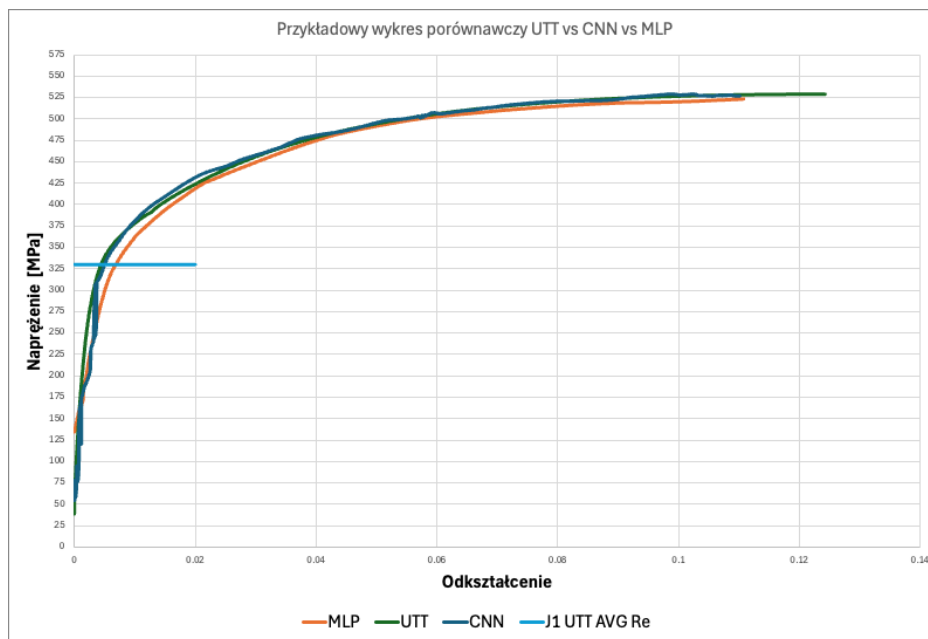


Rysunek 4.32: Wykresy prawdopodobieństwa dla danych z predykcji granicy plastyczności stali 10H2M



Rysunek 4.33: Wykresy prawdopodobieństwa dla danych z predykcji granicy wytrzymałości stali 10H2M

oceny, jak będzie to miało miejsce w praktycznym zastosowaniu. Na rysunku 4.34 zobrazowano wyniki predykcji próbki materiału D1 pochodzącej z testu SPT i oznaczonej *500_24C_42* w zestawieniu z próbką UTT o oznaczeniu *J1_UTT_F16* w bazie danych. W perspektywie całego przebiegu krzywej, przy nałożeniu na siebie wszystkich 3 wykresów, dopasowanie jest dobre. Rozróżnialne dla każdego z modeli jest charakterystyczne zwichrowanie predykcji z modelu CNN, jednak podążanie ścieżką wzorcową UTT oraz gładki przebieg krzywej MLP ma jednak większe nachylenie w zakresie sprężystym.

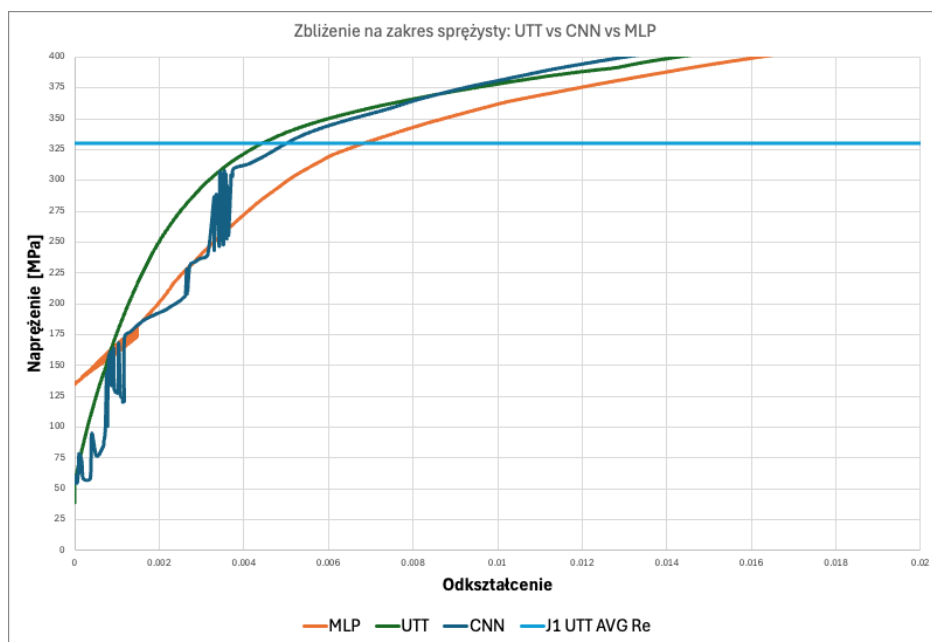


Rysunek 4.34: Przykładowe zestawienie wyników predykcji z rzeczywistą krzywą UTT dla materiału J1

Aby jednak w pełni zrozumieć naturę problemu automatyzacji odczytu wyników z gotowych wykresów i właściwie dobrać *postprocessing*, aby korygować problemy specyficzne dla danej sieci czy nawet konkretnej predykcji, należy przyjrzeć się bliżej przebiegom krzywych w okolicach spodziewanej wartości granicy plastyczności (Rysunek 4.35). Widoczny jest tam trudny przebieg krzywej generowanej z modelu CNN, niemożliwy do pełnego przewidzenia w czasie obróbki danych.

4.2.2 Stal 13HMF

Podobnie jak dla stali 10H2M, wyniki predykcji oraz wyniki porównawcze UTT zestawiono w tabeli (4.14). Od razu można zauważyć bardzo dobre dopasowanie względem granicy wytrzymałości, wynoszące około 1%. Dla tej stali widoczne są również zdecydowanie bardziej zbliżone wartości granicy plastyczności, a może to wynikać z faktu, iż dla obu klas materiału w badaniu UTT występowała wyraźna granica plastyczności, co dawało dużo silniejszy przekaz przy uczeniu się sieci neuronowych niż gładka krzywa materiałowa. Dzięki temu zjawisku i towarzyszącym mu dłuższym rejonom w formie siodła bez wzrostu naprężeń, obie sieci z powodzeniem mogły nauczyć się tej charakterystyki. Jednak wartości odchylenia standardowego dla grupy M3 CNN wskazują, że podejście *Curve to Curve* sprawdza się w takich sytuacjach zdecydowanie lepiej niż *Point to Point*, zapamiętując schematy związane z tym zjawiskiem. Ta sama sieć jednak dla materiału R1 wygenerowała 7 wyników odstających, które zostały statystycznie odrzucone z dalszej analizy,



Rysunek 4.35: Zbliżenie na zakres sprężysty wykresów z predykcji NN

co obniżyło użyteczną efektywność tej grupy. Wyniki predykcji próbek ze zbioru testowego pochodzącego z fazy uczenia sieci przedstawiono w tabeli (4.15). Dla sieci CNN było to po 7 próbek M3 i R1, dla sieci MLP 4 próbki R1 oraz 5 próbek M3. Obie metody predykcji ponownie dają wartości odchylenia od średniej liczonej w kilku procentach, jednak model CNN dla grupy M3 wykazał uśrednione wyniki R_m wyższe niż zakładany cel. Wyjaśnienia ku temu szukać można w konkretnej próbce 500_22C_209 ze zbioru M3 SPT, która w predykcji modelem CNN wskazała granicę wytrzymałości na poziomie 640 MPa. Pojawienie się tej próbki w zbiorze testowym obniżyło również efektywnie wynik wartości R^2 , podawanej w poprzednim rozdziale przy opisie modeli predykcyjnych. Próbka ta odstawała już w badaniu SPT, wskazując nienaturalnie wysokie wyniki ($R_e = 536$ MPa; $R_m = 608$ MPa), co przełożyło się również na odchylenie wyniku dla predykcji, co potwierdziła dodatkowa analiza wyniku tej próbki z użyciem modelu MLP. Ze względu na brak odfiltrowania tej próbki przed budową bazy danych, przyjmuje się pełną losowość podziału grup na uczące, walidacyjne i testowe podczas przygotowania modeli, co skutkuje obniżonym wynikiem efektywności predykcji modelu CNN dla grupy materiału M3, mając jednocześnie na uwadze pozostałe bardzo dobre wyniki predykcji. Zgodnie z założeniami przyjętymi podczas opisu wpływu próbek odstających na pracę sieci neuronowych w przypadku uwzględnienia ich w bazie danych w rozdziale 3, próbka ta miała wpływ na wyniki oceny pracy sieci. Decyzja o konieczności usunięcia lub uwzględnienia tej próbki w przypadku odtwarzania badań pozostaje otwarta.

Analizując dalej opis rozkładów i testów normalności dla stali 13HMF z tabeli (4.16) widzimy, że tym razem wartości z testu UTT są niemalże idealnie w zakresie normalnego rozkładu. Dla wyników z predykcji MLP widzimy skrzywienia, udokumentowane również wyższymi wartościami odchylenia standardowego. Wyjaśnienia tej obserwacji należy szukać ponownie w automatycznej metodzie odczytu granicy plastyczności z wygenerowanych krzywych, a takie wartości, obserwowane jednostkowo, powinny być weryfikowane przez operatora.

Property	Material	Mean [MPa]	Median [MPa]	Std Dev	[%] AVG Δ to UTT	N	Outliers Removed	Usable N [%]
Re	M3 CNN	386	386	3.75	0.5%	36	2	95%
Re	M3 MLP	391	384	20.09	1.8%	38	0	100%
Re	M3 UTT	384	384	7.09	-	28	2	93%
Re	R1 CNN	458	458	4.14	2.2%	25	7	78%
Re	R1 MLP	428	432	10.19	-4.5%	29	3	91%
Re	R1 UTT	448	448	9.66	-	28	1	97%
Rm	M3 CNN	530	530	2.66	1.0%	36	2	95%
Rm	M3 MLP	528	528	3.5	0.6%	37	1	97%
Rm	M3 UTT	525	525	6.41	-	28	2	93%
Rm	R1 CNN	642	641	4.92	0.6%	31	1	97%
Rm	R1 MLP	633	633	3.71	-0.8%	29	3	91%
Rm	R1 UTT	638	636	10.94	-	29	0	100%

Tabela 4.14: Tabela porównawcza dla wyników predykcji dla materiału 13HMF

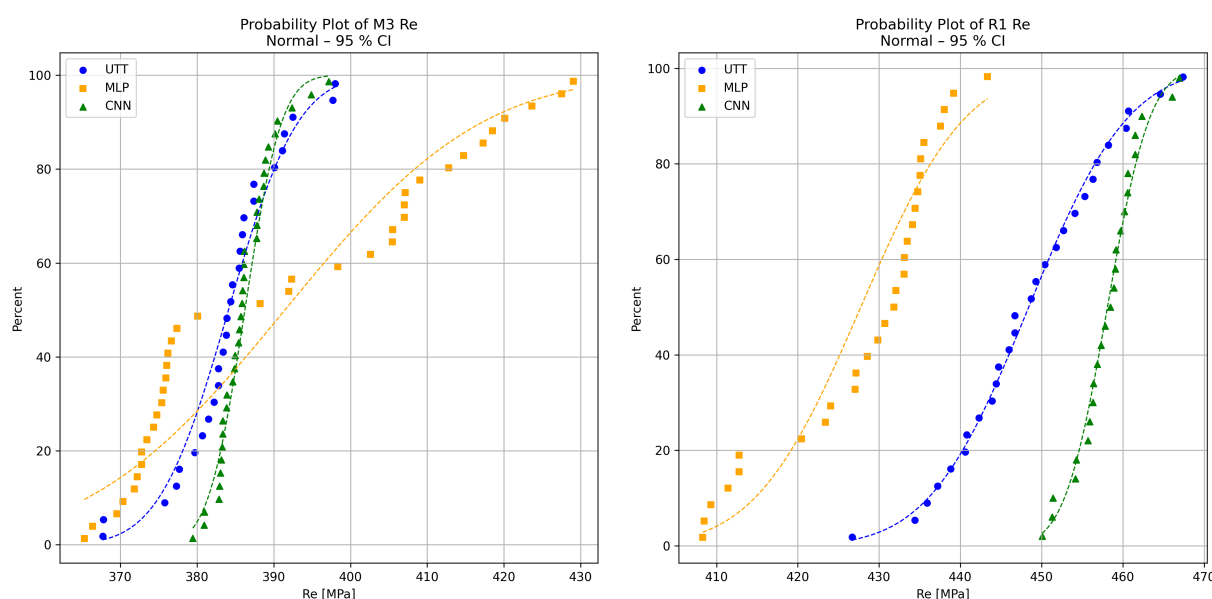
Material	M3				R1			
Property	Re		Rm		Re		Rm	
Unit	[MPa]	Δ [%]	[MPa]	Δ [%]	[MPa]	Δ [%]	[MPa]	Δ [%]
UTT	384	-	525	-	448	-	638	-
CNN	390	1.7%	545	3.7%	439	-1.9%	645	1.1%
MLP	393	2.3%	534	1.7%	426	-4.9%	626	-1.8%

Tabela 4.15: Wyniki walidacji próbek ze zbioru testowego 13HMF

Property	Material	AD Statistic	Shapiro p	Normality	Best-Fit Dist	N
Re	M3 CNN	0.589	0.088	Normal	norm	36
Re	M3 MLP	1.826	0.001	Not Normal	weibull_min	38
Re	M3 UTT	0.493	0.249	Normal	norm	28
Re	R1 CNN	0.234	0.795	Normal	norm	25
Re	R1 MLP	1.414	0.004	Not Normal	weibull_min	29
Re	R1 UTT	0.085	1	Normal	norm	28
Rm	M3 CNN	0.281	0.769	Normal	norm	36
Rm	M3 MLP	0.257	0.904	Normal	norm	37
Rm	M3 UTT	0.238	0.916	Normal	norm	28
Rm	R1 CNN	1.212	0.002	Not Normal	lognorm	31
Rm	R1 MLP	0.302	0.358	Normal	norm	29
Rm	R1 UTT	0.332	0.714	Normal	norm	29

Tabela 4.16: Podsumowanie testów dopasowania normalnego dla stali 13HMF

Wykresy prawdopodobieństwa dla granicy plastyczności materiału 13HMF przedstawiono na rysunku 4.36. Sieć CNN dla grupy M3 obrazuje dokładność predykcji w zakresie granicy plastyczności opisaną wcześniej statystycznie. Odzworowanie przebiegu krzywej naprężenie-odkształcenie w zakresie przejścia ze stanu sprężystego w plastyczny zapewniło łatwość odczytu wartości z wykresu przez algorytm. Sieć MLP jest z kolei widocznie podzielona na dwie grupy danych; grupę ściśle upakowaną o niewielkim przesunięciu rzędu 10 MPa względem dopasowania do UTT oraz grupę szeroko rozłożoną w zakresie 50 MPa. Jest to opisane wysokim odchyleniem standardowym dla tego zbioru danych. W przypadku grupy R1 sieć MLP ponownie wykazała się rozkładem nienormalnym, opisanym w tabeli (4.16) jako rozkład minimalny Weibulla z wyraźnym ogonem danych oddzielającym się od reszty. Mimo tych obserwacji odchyłka średnia dla sieci MLP dla tego parametru wynosi poniżej 5%.

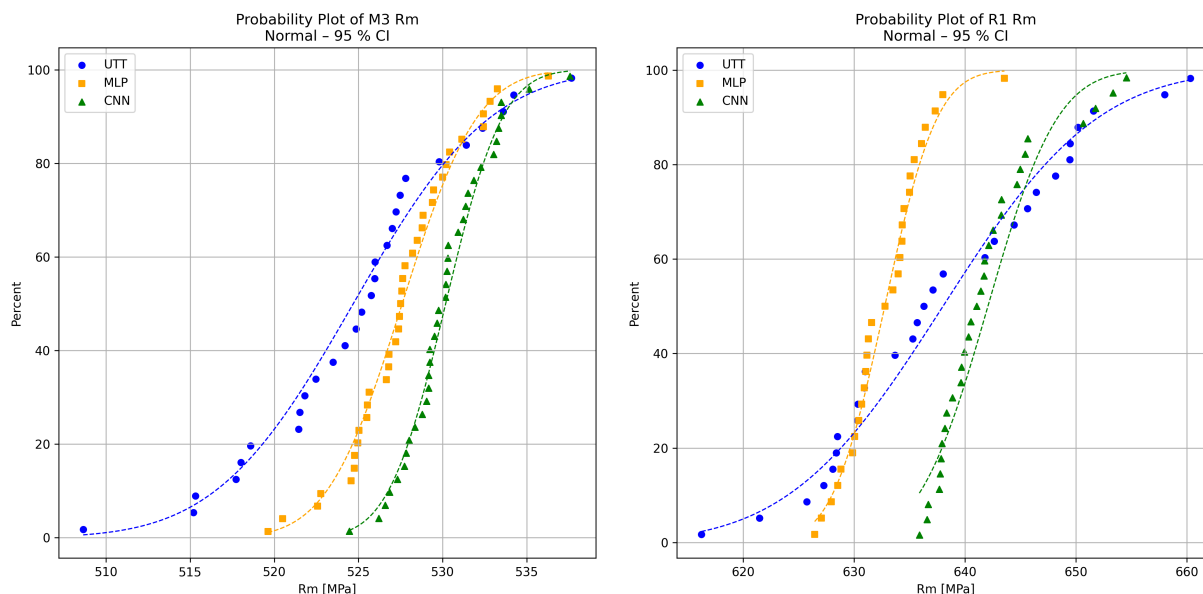


Rysunek 4.36: Wykresy prawdopodobieństwa dla danych z predykcji granicy plastyczności stali 13HMF

Rozkłady wartości dla granicy wytrzymałości na wykresach (4.37). Porównując wyniki UTT pomiędzy obydwooma stanami materiału 13HMF, można zauważyć podobieństwo w zachowaniu populacji danych. Mimo klasyfikacji jako rozkłady normalne, ze względu na symetrię, widoczne są grupy punktów skrajnych oraz podział na mniejsze podgrupy. Takie zachowanie wyników stanowiło przeszkodę dla nauczania sieci neuronowej. Jak widać, w przypadku materiału R1 decydowało to o wycentrowaniu wyników predykcji danej sieci, zależnie od wagi przypisanej do danej grupy próbek UTT w czasie uczenia. Dla grupy M3 wyniki, mimo przeszacowania względem wzoru w postaci UTT, są zgodne z faktycznym stanem materiału. Należy podkreślić, że dla obu grup materiałów i obu sieci średnie wyniki predykcji mieszczą się w granicy błędu 1% względem wartości średniej populacji UTT.

4.2.3 Stal 15HM

Stal 15HM jest ostatnim materiałem analizowanym pod kątem działania i predykcji sieci neuronowych. Wyniki przedstawione w tabeli (4.17). Ponownie utrzymującym się trendem jest wysoka dokładność pre-



Rysunek 4.37: Wykresy prawdopodobieństwa dla danych z predykcji granicy wytrzymałości stali 13HMF

dykcji granicy wytrzymałości obu stanów materiału dla obu sieci. Pojawia się jednak widoczny problem w predykcji granicy plastyczności dla grupy T1. Żadna z sieci nie doszacowuje wartości, zapewniając jednak dużą powtarzalność, zobrazowaną niskim odchyleniem standardowym. Weryfikacja źródeł wyników UTT wskazała, że średnia wartość całej populacji w pierwotnym raporcie wyników wynosiła 406.5 MPa, co jest wartością znacznie bliższą predykcjom sieci neuronowych. Zwraca to uwagę na 5 próbek z grupy Re T1 UTT sklasyfikowanych jako odstające zgodnie z podejściem opisanym w rozdziale 3, w sekcji 3.4.2, które mogły zaniżać wagę predykcji dla tego fragmentu krzywej dla materiału T1.

Property	Material	Mean [MPa]	Median [MPa]	Std Dev	[%] AVG Δ to UTT	N	Outliers Removed	Usable N [%]
Re	H1 CNN	364	364	2.12	-0.8%	33	2	94%
Re	H1 MLP	356	358	5.61	-3.0%	30	5	86%
Re	H1 UTT	367	369	13.46	-	28	2	93%
Re	T1 CNN	401	400	5.95	-4.1%	32	1	97%
Re	T1 MLP	401	402	6.99	-4.1%	30	3	91%
Re	T1 UTT	418	425	21.24	-	25	5	83%
Rm	H1 CNN	482	481	1.6	-1.4%	33	2	94%
Rm	H1 MLP	482	481	3.82	-1.4%	34	1	97%
Rm	H1 UTT	489	490	10.05	-	28	2	93%
Rm	T1 CNN	548	547	3.29	1.3%	31	2	94%
Rm	T1 MLP	542	542	2.11	0.2%	31	2	94%
Rm	T1 UTT	541	545	14.95	-	30	0	100%

Tabela 4.17: Tabela porównawcza dla wyników predykcji dla materiału 15HM

Dla ostatniego z materiałów wyniki testów na zbiorach testowych z uczenia modeli sieci neuronowych przedstawiono w tabeli (4.18), na podstawie 7 próbek z każdej grupy dla sieci CNN i 3 próbek dla sieci MLP. Obie sieci neuronowe powinny wskazywać z dobrą dokładnością wartości granicy wytrzymałości materiałów. W przypadku obu sieci schemat niedoszacowania granicy plastyczności materiału T1 utrzymuje się zgodnie z trendem całej populacji. Dla materiału H1 wysoka precyzja i dokładność podkreślają wynik modelu CNN.

Material	H1				T1			
Property	Re		Rm		Re		Rm	
Unit	[MPa]	Δ [%]	[MPa]	Δ [%]	[MPa]	Δ [%]	[MPa]	Δ [%]
UTT	367	-	489	-	418	-	541	-
CNN	365	-0.5%	479	-2.0%	399	-4.5%	547	1.1%
MLP	345	-6.0%	478	-2.2%	400	-4.3%	544	0.6%

Tabela 4.18: Wyniki walidacji próbek ze zbioru testowego 15HM

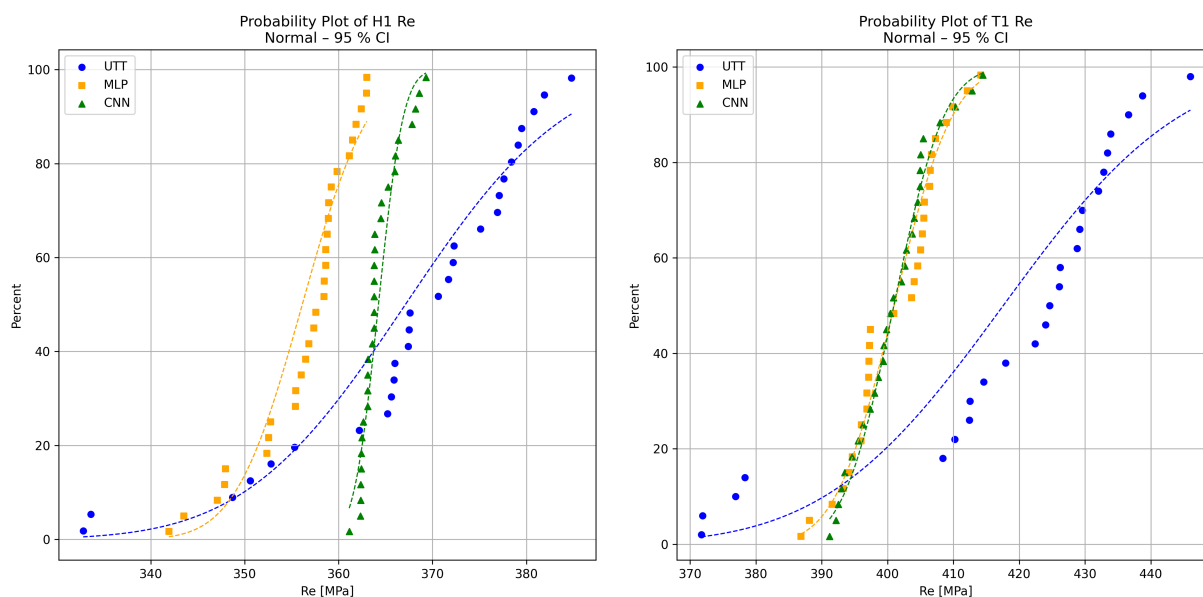
Tabela (4.19) przedstawia, jak poszczególne grupy i własności materiałowe dla nich opisane dla materiału 15HM zachowują się pod kątem statystycznego rozkładu danych. Wyróżniającą się grupą jest T1 dla wartości Re, gdzie mimo przesunięcia względem oczekiwanych wartości, obie sieci prezentują rozkład normalny, podparty opisaniem niskim odchyleniem standardowym, co obrazuje precyzję wyników, przyjmując wspomniane 4% odchylenie od statystycznego celu.

Property	Material	AD Statistic	Shapiro p	Normality	Best-Fit Dist	N
Re	H1 CNN	0.998	0.035	Not Normal	lognorm	33
Re	H1 MLP	1.252	0.004	Not Normal	weibull_min	30
Re	H1 UTT	1.014	0.006	Not Normal	weibull_min	28
Re	T1 CNN	0.261	0.442	Normal	norm	32
Re	T1 MLP	0.591	0.334	Normal	norm	30
Re	T1 UTT	1.61	0.001	Not Normal	lognorm	25
Rm	H1 CNN	0.735	0.043	Not Normal	lognorm	33
Rm	H1 MLP	0.846	0.008	Not Normal	weibull_min	34
Rm	H1 UTT	0.564	0.113	Normal	norm	28
Rm	T1 CNN	1.463	0.001	Not Normal	lognorm	31
Rm	T1 MLP	0.594	0.182	Normal	norm	31
Rm	T1 UTT	0.812	0.051	Normal	norm	30

Tabela 4.19: Podsumowanie testów dopasowania normalnego dla stali 15HM

Wykresy prawdopodobieństwa granicy plastyczności z predykcji dla stali 15HM przedstawiono na wykresach (4.38) pokazują bardzo dobre dopasowanie w grupie H1 do wzoru UTT wartości wygenerowanych przez obie sieci neuronowe, a następnie odczytane z wykresów, mimo szerokiego zakresu rozkładu danych UTT w charakterze rozkładu minimalnego Weibulla. Dla grupy T1 widoczne jest opisane wcześniej stałe

przesunięcie w postaci niedoszacowania wartości na wykresie w momencie przejścia materiału w stan plastyczny dla wyników predykcji. Obie sieci wykazują jednak doskonałe dopasowanie wartości względem siebie, co wskazuje na dobry proces uczenia i stabilne wyniki zorientowane. Rozkład wyników predyk-

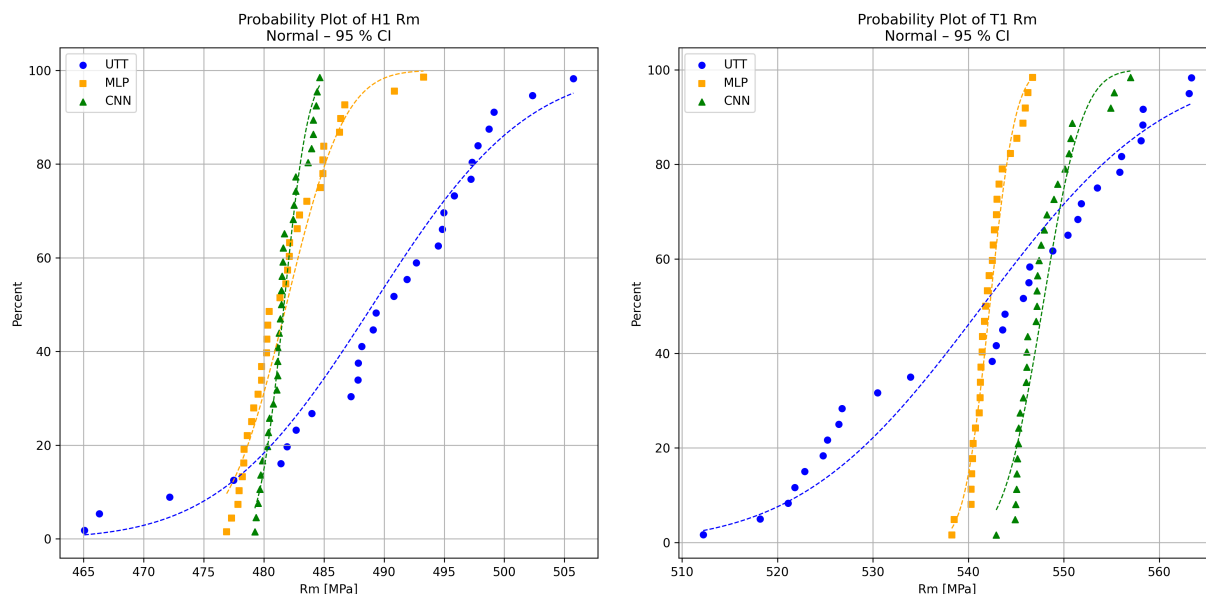


Rysunek 4.38: Wykresy prawdopodobieństwa dla danych z predykcji granicy plastyczności stali 15HM

cji granicy wytrzymałości przedstawiony jest na wykresach prawdopodobieństwa (4.39) dla stali 15HM. Pierwszą obserwacją jest niemalże identyczne dopasowanie predykcji w grupie H1 z obu sieci neuronowych, mimo niewielkiego przesunięcia względem celu, jakim jest średnia wartość UTT. Wyniki z testu klasycznego rozłożone są na szerokości 40 MPa z niewielkimi ogonami w zakresach wartości skrajnych. Dla grupy t1 widoczny jest niewielki rozstrzał wyników, jednak wciąż w granicach prawidłowości predykcji. Wynika on prawdopodobnie z wagi nadal konkretnym wynikiem UTT, które są wyraźnie podzielone na 2 grupy, z większą w wyższych wartościach naprężenia, co wpływało na przesunięcie wartości predykcji w tę stronę.

4.2.4 Podsumowanie wyników predykcji

Jako kryterium akceptacji wyników predykcji z wykorzystaniem sieci neuronowych postawiono cel, aby wyniki własności materiałów badanych i analizowanych były bliższe wartościom otrzymywanym z testu UTT niż wyniki bazowe z SPT. W niniejszym rozdziale przedstawiono ogólne zestawienie dla danego materiału, a w tabelach 4.20 i 4.21 porównano te wyniki w kontekście przyjętego kryterium. W ostatniej kolumnie umieszczono średni bezwzględny wynik odchylenia z określonej metody dla badanego parametru. Zwracając najpierw uwagę na wyniki zestawienia dla granicy plastyczności, w grupie D1, stanowiącej odzwierciedlenie stanu nowej stali 10H2M przed eksploatacją, żadna z metod nie przedstawiła zadowalającego wyniku. W rozumieniu badania charakterystyka SPT sprawia, że jest to wynik wiarygodny dla samego badania, jednak wyniki predykcji z sieci neuronowych mogą zostać poddane dodatkowej weryfikacji otrzymanych wykresów w celu manualnej oceny wyników oraz potwierdzenia lub odrzucenia oceny automatycznej



Rysunek 4.39: Wykresy prawdopodobieństwa dla danych z predykcji granicy wytrzymałości stali 15HM

Odchylenie średniej wyników Re ze wszystkich źródeł względem wyników UTT [%]							
Grupa	J1	D1	M3	R1	H1	T1	ABS AVG
SPT	-5.8	9.5	-7.0	-2.5	-16.6	-1.9	7.2
CNN	2.1	-7.8	0.5	2.2	-0.8	-4.1	2.9
MLP	-0.6	-11.2	1.8	-4.5	-3.0	-4.1	4.2

Tabela 4.20: Odchylenie średniej wyników Re ze wszystkich źródeł względem wyników UTT

ze względu na trudności w interpretacji algorytmicznego, dokładnego wyznaczania umownej granicy plastyczności. Kolejnym wynikiem, na który należy zwrócić uwagę, jest ocena granicy plastyczności materiału T1 stali 15HM, tj. wariantu materiału przepracowanego. Jak wspomniano w poprzedniej podsekcji, wyniki przyjmuje się za zgodne ze względu na wysokie rozstrzelenie wyników T1, które powoduje statystyczne odrzucenie próbek odstających przed analizą populacji. W kontekście największej dokładności oceny stanu materiału wyniki predykcji z sieci neuronowej typu CNN wykazały najmniejsze odchylenie od UTT.

Dla granicy wytrzymałości zauważyć można, że badanie SPT przeszacowuje wyniki względem UTT. Opisano to podczas dokładnej analizy populacji wyników, gdzie zauważono tendencję do wyników o 20-30 MPa wyższych. Uśredniona ocena wyników wskazuje na dokładność SPT nieco poniżej 5%. W przypadku predykcji przez sieci neuronowe obie architektury modeli wykazały bardzo dobrą dokładność wyników.

Odchylenie średniej wyników Rm ze wszystkich źródeł względem wyników UTT [%]							
Grupa	J1	D1	M3	R1	H1	T1	ABS AVG
SPT	3.0	2.9	6.3	1.9	5.7	9.6	4.9
CNN	-0.6	0.6	1.0	0.6	-1.4	1.3	0.9
MLP	-1.7	-0.8	0.6	-0.8	-1.4	0.2	0.9

Tabela 4.21: Odchylenie średniej wyników Rm ze wszystkich źródeł względem wyników UTT

Rozdział 5

Podsumowanie i wnioski

5.1 Dyskusja i interpretacja wyników w kontekście celu pracy

W niniejszej pracy podjęto próby usprawnienia analizy materiałowej próbek pobranych z wyeksploatowanych konstrukcji pracujących w trudnych warunkach, stosując do tego celu metody SPT. Przeprowadzono szereg badań fizycznych oraz analiz komputerowych mających na celu przybliżenie zagadnienia, jakim jest automatyzacja oceny stanu materiału z wykorzystaniem metody SPT. Przygotowano i przeanalizowano łącznie ponad 400 próbek SPT i 180 próbek UTT. Postawiono tezę, że SPT jest metodą konkurencyjną dla klasycznych metod, takich jak próba jednoosiowego rozciągania, co pozwala na skuteczne zastąpienie ich w konkretnych warunkach, kiedy niemożliwe jest pobranie pełnowymiarowej próbki standardowej. SPT klasyfikowane jest jako metoda nieniszcząca, co daje jej znaczącą przewagę w zastosowaniach takich jak stały monitoring konstrukcji i pozwala na wykorzystanie równolegle do wspierających się, ugruntowanych już metod NDT. Korzyści wynikające z zastosowania sieci neuronowych wspomagają pełną ocenę materiału i pozwalają na otrzymanie wartości spodziewanych ze standardowej próby jednoosiowego rozciągania.

Przygotowane sieci neuronowe dla wszystkich 3 stali osiągnęły zakładane progi dokładności dla obydwu parametrów będących tematem dyskusji: granicy plastyczności i granicy wytrzymałości. Kryterium akceptacji wyników było przedstawienie zestawów danych wygenerowanych przez sieć neuronową na podstawie surowych danych wynikowych ze średnią dokładnością predykcji wyniku wyższą niż wartości granicy plastyczności i granicy wytrzymałości w metodzie SPT, wyznaczone z wzorów podawanych w literaturze. Stosowanie metody nieniszczącej u samych podstaw jest korzystniejsze finansowo od metod wymagających wyłączenia, nawet tymczasowego, badanej konstrukcji z użytku. W połączeniu ze wspomaganiem komputerowo wyznaczaniem właściwości materiałowych, SPT klasyfikuje się jako metodę atrakcyjną do komercyjnych i przemysłowych zastosowań.

Na podstawie przeprowadzonych analiz proponuje się, aby do dalszych zastosowań korzystać z predykcji za pomocą sieci typu 1D CNN, która wykazała się większą dokładnością predykcji granicy plastyczności, wskazując na dokładniejsze odwzorowanie przebiegu krzywej materiałowej, ale również z powodu możliwych potencjalnych usprawnień i poszerzania możliwości tej sieci.

Pobieranie próbek w laboratorium i przeprowadzanie ich przez całościowy proces od bloku materiału surowego do wyniku oceny własności i predykcji względem UTT nie zostało wcześniej udokumentowane w

dostępnej literaturze ani zaimplementowane w praktyce, co stanowiło cel użyteczny pracy. Wyniki przedstawione w poprzednim rozdziale pozwalają na bezpośrednie usprawnienie procesu badania z wykorzystaniem metody SPT. Dzięki przygotowanym modelom otrzymane wyniki można dodatkowo przeanalizować pod kątem weryfikacji z UTT, a poznane wartości mogą pomóc w dalszej ocenie. Jak podkreślano w pracy, SPT wykazuje się zdecydowanie wyższą czułością na potencjalne zmiany występujące w materiale. Pozwala to na ocenę wyników pod różnymi kątami. Kompleksowe wyniki z predykcji mogą służyć jako odniesienie do spodziewanego stanu przy analizie UTT, a wyniki SPT mogą być z nią dalej porównywane. W kontekście istniejących prac otrzymane wyniki są w zgodności z podobnymi próbami bezpośredniej predykcji wartości skalarnych parametrów materiałowych na podstawie danych wejściowych SPT oraz prób symulacji krzywych materiałowych dla innych materiałów niemetalicznych na podstawie danych pochodzących z symulacji MES.

Badania literaturowe przeprowadzone na wczesnym etapie prac wykazały, jakie dotychczas stosowano metody korelacji i predykcji wyników SPT z wynikami UTT. Na tej podstawie przygotowano cel pracy, mający na celu wypełnienie luki istniejącej na polu badań nad predykcją wyników UTT na podstawie badań SPT dla stali wykorzystywanych w przemyśle. Osiągnięcia opisane i udokumentowane w niniejszej pracy wyróżniają się na tle dotychczasowych badań, mimo że bazują na różnych podejściach, które były już częściowo stosowane, i można podsumować je jako:

- **Przygotowanie kompletnego algorytmu analizy danych** - system przechodzi od surowego wyniku z testu SPT do prezentacji kompletnych wykresów naprężenie-odkształcenie z automatyczną oceną własności materiału lub pozwalającą na samodzielną ocenę ekspercką przez operatora. Algorytm jest gotowy do implementacji na stanowiskach testowych wykonujących badanie SPT.
- **Budowa sieci neuronowych z wykorzystaniem dekodów wyspecjalizowanych w danym materiale i jego stanach** - takie podejście nie było dotąd stosowane w opublikowanych pracach i może stanowić nowy trend. Jest to również skalowalne rozwiązanie, pozwalające na dalsze douczanie modelu po przygotowaniu danych z nowych, nieznanych jeszcze materiałów, poszerzając jego możliwości. Szczególnie istotne jest, kiedy dekodery specjalizuje się we wszystkich stanach danego materiału.
- **Pionierskie modelowanie *Curve to Curve*** - wykorzystujące przejście z pełnej krzywej SPT w postaci surowych danych do pełnej krzywej UTT, pozwala na dowolną lub automatyczną interpretację wyników, w zależności od potrzeb. Rozwiązanie to nie było wcześniej stosowane do predykcji zmian w stanach materiałów na podstawie danych eksperymentalnych z rzeczywistych konstrukcji.
- **Wykorzystanie wyłącznie danych eksperymentalnych** - podejście to pozwoliło odwzorować rzeczywiste zachowania materiałów w różnych stanach eksploatacji, przybliżając możliwość aplikacji modelu w rzeczywistych warunkach.

5.2 Ograniczenia, potencjalne usprawnienia i rekomendacje

Zagadnienia, jakimi jest automatyzacja oceny stanu materiału, są szeroko pojętym celem, z którym borykają się badacze na całym świecie. Metody wykorzystywane w próbach przedstawianych w wielu pracach przez

wiele lat wskazują na trend, który polega na chęci przygotowania perfekcyjnego modelu predykcyjnego. Niniejsza praca jest pionierska w tej dziedzinie dzięki zaproponowanym i zweryfikowanym rozwiązaniom implementacji sieci neuronowych do oceny własności mechanicznych materiałów na podstawie badania SPT małych próbek. Pozostaje jednak szerokie pole do przyszłych badań zajmujących się Small Punch Test, a do propozycji usprawnień wzajemnie się komplementujących należą:

- Poszerzenie bazy danych o większą liczbę materiałów w szerszej gamie stanów eksploatacji. Pozwoli to na pokrycie większej powierzchni rynku przy komercyjnych zastosowaniach, np. przez Urząd Dozoru technicznego.
- Wdrożenie w pełni automatycznych klasyfikatorów dla stanów eksploatacji na podstawie krzywych SPT. Klasyfikatory nauczone na próbkach ze stanów pośrednich mogą pomóc w nakierowaniu dekodów modelu sieci neuronowej w celu dokładnej predykcji własności danego materiału.
- Poszerzenie zakresu dekodów materiałowych o konkretne stany eksploatacji w celu usprawnienia predykcji w zakresie danego materiału. W połączeniu ze wspomnianymi klasyfikatorami sieć może stanowić kompleksowe narzędzie do analizy większości stali stosowanych w danym segmencie przemysłu.
- Rozwój predykcji innych pobocznych własności materiału, takich jak na przykład wartość procentowego wydłużenia, znana z UTT, poprzez wsparcie modelu o pracę na obrazach w postaci zdjęć mikroskopowych próbek. W literaturze istnieje luka w postaci braku potwierdzenia metod oceny wydłużenia znanego z UTT z wykorzystaniem SPT. Dodanie wizualnych danych połączonych z danymi numerycznymi mogłoby pomóc w osiągnięciu tego celu.

Dodatek A

Kod źródłowy

```
1 import os
2 import csv
3 import pandas as pd
4
5 def log(message):
6     from datetime import datetime
7     timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
8     print(f"{timestamp} - {message}")
9
10 def remove_header_and_quotes(file_path):
11     with open(file_path, 'r') as f:
12         lines = f.readlines()
13         cleaned_lines = [line.replace('"', '') for line in lines[5:]]
14         return cleaned_lines
15
16 def remove_extra_columns_and_set_header(lines):
17     reader = csv.reader(lines, delimiter='\t')
18     rows = [row for row in reader]
19     if not rows or len(rows[0]) < 2:
20         return None
21     rows[0] = ["Distance", "Force"]
22     rows = [[row[0], row[1]] for row in rows]
23     return rows
24
25 def reorder_columns(rows):
26     for i in range(len(rows)):
27         if len(rows[i]) >= 2:
28             rows[i][0], rows[i][1] = rows[i][1], rows[i][0]
29     return rows
30
31 def trim_data(rows):
32     header = rows[0]
33     data_rows = rows[1:]
34     df = pd.DataFrame(data_rows, columns=header)
35     df['Force'] = df['Force'].str.replace(',', '.').astype(float)
```

```

36     max_force_index = df['Force'].idxmax()
37     trimmed_df = df.iloc[:max_force_index + 1]
38     return trimmed_df
39
40 def save_final_file(df, original_file_path):
41     output_file_path = os.path.join(os.path.dirname(original_file_path), "processed_"
42     + os.path.basename(original_file_path))
43     df.to_csv(output_file_path, index=False, sep='\t', decimal=',')
44     log(f"Processed file saved as: {output_file_path}")
45
46 def process_file(file_path):
47     log(f"Processing file: {file_path}")
48     cleaned_lines = remove_header_and_quotes(file_path)
49     rows = remove_extra_columns_and_set_header(cleaned_lines)
50     if rows is None:
51         log(f"Skipping {file_path} due to insufficient columns.")
52         return
53     rows = reorder_columns(rows)
54     final_df = trim_data(rows)
55     save_final_file(final_df, file_path)
56
57 def main():
58     root_directory = input("Enter the root directory path: ").strip()
59     for dirpath, _, filenames in os.walk(root_directory):
60         for filename in filenames:
61             if filename.upper().endswith('.TXT'):
62                 file_path = os.path.join(dirpath, filename)
63                 process_file(file_path)
64     log("All files processed successfully.")
65
66 if __name__ == "__main__":
67     main()

```

Listing A.1: Przygotowanie treści plików danych do normalizacji.

```

1 import pandas as pd
2 import os
3
4 def normalize_data(file_path, output_folder):
5     df = pd.read_csv(file_path, sep='\t', decimal=',', usecols=['Force', 'Distance'])
6
7     df['Force'] = df['Force'].clip(lower=0)
8     df['Distance'] = df['Distance'].clip(lower=0)
9
10    normalized_force = df['Force'] / # 13500.0 dla UTT oraz 1800.0 dla SPT
11    normalized_distance = df['Distance'] / # 2.7 dla UTT oraz 1.8 dla SPT
12
13    normalized_df = pd.DataFrame({
14        'Normalized Force': normalized_force,

```

```

15         'Normalized Distance': normalized_distance
16     })
17
18     if not os.path.exists(output_folder):
19         os.makedirs(output_folder)
20
21     output_file_path = os.path.join(output_folder, 'normalized_' + os.path.basename(
22         file_path))
23     normalized_df.to_csv(output_file_path, index=False, decimal=',', sep='\t')
24
25 def process_all_files(source_root, output_root):
26     for dirpath, dirnames, filenames in os.walk(source_root):
27         for filename in filenames:
28             if filename.endswith('.txt'):
29                 file_path = os.path.join(dirpath, filename)
30
31                 relative_path = os.path.relpath(dirpath, source_root)
32                 output_folder = os.path.join(output_root, relative_path)
33
34                 normalize_data(file_path, output_folder)
35                 print(f'Processed file: {filename}')
36
37 source_root = '/Users/...'
38 output_root = '/Users/...'
39
40 process_all_files(source_root, output_root)

```

Listing A.2: Normalizacja danych.

```

1 import pandas as pd
2 import sqlite3
3 import os
4 from datetime import datetime
5
6 def log(message):
7     timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
8     print(f"{timestamp} - {message}")
9
10 def convert_and_clean_data(df):
11     for col in ['Normalized Force', 'Normalized Distance']:
12         df[col] = df[col].astype(str).str.replace(',', '.').astype('float32')
13     return df
14
15 def pair_data_based_on_percentiles(spt_data, utt_data, sample_rate=10):
16     paired_data = []
17
18     spt_data = convert_and_clean_data(spt_data)
19     utt_data = convert_and_clean_data(utt_data)

```

```

20
21 sampled_spt_data = spt_data.iloc[:, :sample_rate]
22 num_rows_sampled_spt = len(sampled_spt_data)
23 num_rows Utt = len(utt_data)
24
25 for index in range(num_rows_sampled_spt):
26     corresponding_Utt_index = int(index * num_rows_Utt / num_rows_sampled_spt)
27
28     paired_row = {
29         'SPT_Force': sampled_spt_data.iloc[index]['Normalized Force'],
30         'SPT_Distance': sampled_spt_data.iloc[index]['Normalized Distance'],
31         'Utt_Force': utt_data.iloc[corresponding_Utt_index]['Normalized Force'],
32         'Utt_Distance': utt_data.iloc[corresponding_Utt_index]['Normalized
Distance']
33     }
34     paired_data.append(paired_row)
35
36 return pd.DataFrame(paired_data)
37
38 def insert_data_into_db(paired_data, spt_file, Utt_file, material, cursor):
39     for index, row in paired_data.iterrows():
40         cursor.execute("""
41             INSERT INTO paired_data (Material, SPT_File, Utt_File, SPT_Force,
SPT_Distance, Utt_Force, Utt_Distance)
42             VALUES (?, ?, ?, ?, ?, ?, ?)
43             """, (material, spt_file, Utt_file,
44                 row['SPT_Force'], row['SPT_Distance'],
45                 row['Utt_Force'], row['Utt_Distance']))
46
47 def process_and_pair_folders(spt_folder, Utt_folder, material, cursor):
48     spt_files = [f for f in os.listdir(spt_folder) if f.endswith('.txt')]
49     Utt_files = [f for f in os.listdir(Utt_folder) if f.endswith('.txt')]
50
51     def safe_read_csv(file_path, sep='\t', encoding='utf-8'):
52         try:
53             return pd.read_csv(file_path, sep=sep, encoding=encoding)
54         except UnicodeDecodeError:
55             # Attempt to read with ASCII encoding if UTF-8 fails
56             return pd.read_csv(file_path, sep=sep, encoding='ascii')
57
58     for spt_file in spt_files:
59         spt_data = safe_read_csv(os.path.join(spt_folder, spt_file))
60
61         for Utt_file in Utt_files:
62             Utt_data = safe_read_csv(os.path.join(Utt_folder, Utt_file))
63
64             # Pair the data based on percentiles
65             paired_data = pair_data_based_on_percentiles(spt_data, Utt_data)
66

```

```

67         # Insert the paired data into the database
68         insert_data_into_db(paired_data, spt_file, utt_file, material, cursor)
69
70
71 conn = sqlite3.connect('material_data.db')
72 cur = conn.cursor()
73 cur.execute('DROP TABLE IF EXISTS paired_data')
74 cur.execute('''
75     CREATE TABLE IF NOT EXISTS paired_data (
76         ID INTEGER PRIMARY KEY,
77         Material TEXT,
78         SPT_File TEXT,
79         UTT_File TEXT,
80         SPT_Force REAL,
81         SPT_Distance REAL,
82         UTT_Force REAL,
83         UTT_Distance REAL
84     )
85 ''')
86
87 spt_root = '/Users/...'
88 utt_root = '/Users/...'
89 materials = ['D1', 'H1', 'J1', 'M3', 'R1', 'T1']
90
91 for material in materials:
92     spt_folder = os.path.join(spt_root, material + ' SPT')
93     utt_folder = os.path.join(utt_root, material + ' UTT')
94     log(f"Processing material: {material}")
95     process_and_pair_folders(spt_folder, utt_folder, material, cur)
96     log(f"Completed processing material: {material}")
97
98 conn.commit()
99 conn.close()
100 log("All materials processed and database updated.")

```

Listing A.3: Konstrukcja bazy danych: parowanie i agregacja

```

1 import torch
2 import torch.nn as nn
3 import torch.nn.functional as F
4 import torch.optim as optim
5 import sqlite3
6 import pandas as pd
7 from sklearn.model_selection import train_test_split
8 from torch.utils.data import DataLoader, TensorDataset
9 import optuna
10 import os
11
12

```

```

13 def log(message):
14     from datetime import datetime
15     timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
16     print(f"{timestamp} - {message}")
17
18
19 class MyNeuralNetwork(nn.Module):
20     def __init__(self, hidden_layer_size, activation_function, dropout_rate=0.0,
21                 negative_slope=0.01):
22         super(MyNeuralNetwork, self).__init__()
23         self.fc1 = nn.Linear(2, hidden_layer_size)
24         self.fc2 = nn.Linear(hidden_layer_size, hidden_layer_size)
25         self.fc3 = nn.Linear(hidden_layer_size, hidden_layer_size)
26         self.fc4 = nn.Linear(hidden_layer_size, 2)
27
28         self.activation_function = activation_function
29
30         if activation_function == "Dropout":
31             self.dropout1 = nn.Dropout(dropout_rate)
32             self.dropout2 = nn.Dropout(dropout_rate)
33             self.dropout3 = nn.Dropout(dropout_rate)
34
35         self.leaky_relu = nn.LeakyReLU(negative_slope)
36
37     def forward(self, x):
38         if self.activation_function == "LeakyReLU":
39             x = self.leaky_relu(self.fc1(x))
40             x = self.leaky_relu(self.fc2(x))
41             x = self.leaky_relu(self.fc3(x))
42         else:
43             x = F.relu(self.dropout1(self.fc1(x)))
44             x = F.relu(self.dropout2(self.fc2(x)))
45             x = F.relu(self.dropout3(self.fc3(x)))
46
47         return self.fc4(x)
48
49 def create_data_loaders(batch_size):
50     database_path = '/Users/.../material_data.db'
51     conn = sqlite3.connect(database_path)
52     query = "SELECT Material, SPT_Force, SPT_Distance, UTT_Force, UTT_Distance FROM
53             paired_data"
54     data = pd.read_sql_query(query, conn)
55     conn.close()
56
57     features = data[['SPT_Force', 'SPT_Distance']].values
58     targets = data[['UTT_Force', 'UTT_Distance']].values
59
60     features_tensor = torch.tensor(features, dtype=torch.float32)

```

```
60     targets_tensor = torch.tensor(targets, dtype=torch.float32)
61
62     train_features, test_features, train_targets, test_targets = train_test_split(
63         features_tensor, targets_tensor,
64         test_size=0.2, random_state=42)
65
66     val_features, test_features, val_targets, test_targets = train_test_split(
67         test_features, test_targets,
68         test_size=0.5, random_state=42)
69
70     train_dataset = TensorDataset(train_features, train_targets)
71     val_dataset = TensorDataset(val_features, val_targets)
72     test_dataset = TensorDataset(test_features, test_targets)
73
74     train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
75     val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=False)
76     test_loader = DataLoader(test_dataset, batch_size=batch_size, shuffle=False)
77     return train_loader, val_loader, test_loader
78
79 def objective(trial):
80     log(f"Starting trial: {trial.number}")
81     lr = trial.suggest_float("lr", 1e-5, 1e-1, log=True)
82     hidden_layer_size = trial.suggest_int("hidden_layer_size", 16, 256, step=16)
83     batch_size = trial.suggest_categorical("batch_size", [32, 64, 128, 512])
84     activation_function = trial.suggest_categorical("activation_function", ["LeakyReLU", "Dropout"])
85     dropout_rate = trial.suggest_float("dropout_rate", 0.1, 0.5)
86     negative_slope = trial.suggest_float("negative_slope", 0.01, 0.2)
87
88     model = MyNeuralNetwork(
89         hidden_layer_size,
90         activation_function,
91         dropout_rate=dropout_rate,
92         negative_slope=negative_slope
93     )
94
95     optimizer = optim.Adam(model.parameters(), lr=lr)
96     criterion = nn.MSELoss()
97     train_loader, val_loader, _ = create_data_loaders(batch_size)
98
99     for epoch in range(5):
100         model.train()
101         for batch in train_loader:
102             inputs, targets = batch
103             optimizer.zero_grad()
104             outputs = model(inputs)
105             loss = criterion(outputs, targets)
```

```
104         loss.backward()
105         optimizer.step()
106
107     model.eval()
108     val_loss = 0.0
109     with torch.no_grad():
110         for batch in val_loader:
111             inputs, targets = batch
112             outputs = model(inputs)
113             loss = criterion(outputs, targets)
114             val_loss += loss.item()
115
116     log(f"Trial {trial.number} completed with val_loss: {val_loss / len(val_loader):.4f}")
117     return val_loss / len(val_loader)
118
119
120 database_path = '/Users/...'
121
122 study = optuna.create_study(direction="minimize")
123 study.optimize(objective, n_trials=100)
124
125 log("Best trial:")
126 trial = study.best_trial
127 for key, value in trial.params.items():
128     log(f"    {key}: {value}")
129
130 best_batch_size = trial.params["batch_size"]
131 best_hidden_layer_size = trial.params["hidden_layer_size"]
132 best_activation_function = trial.params["activation_function"]
133 best_dropout_rate = trial.params.get("dropout_rate", 0.0)
134 best_negative_slope = trial.params.get("negative_slope", 0.01)
135 best_lr = trial.params["lr"]
136
137 model = MyNeuralNetwork(
138     best_hidden_layer_size,
139     best_activation_function,
140     dropout_rate=best_dropout_rate,
141     negative_slope=best_negative_slope
142 )
143
144 optimizer = optim.Adam(model.parameters(), lr=best_lr)
145 criterion = nn.MSELoss()
146 train_loader, val_loader, test_loader = create_data_loaders(best_batch_size)
147
148 log(f"Starting final training with best parameters.")
149 for epoch in range(10):
150     epoch_loss = 0.0
151     for batch in train_loader:
```



```
152     inputs, targets = batch
153     optimizer.zero_grad()
154     outputs = model(inputs)
155     loss = criterion(outputs, targets)
156     loss.backward()
157     optimizer.step()
158     epoch_loss += loss.item()
159     average_loss = epoch_loss / len(train_loader)
160     log(f'Epoch [{epoch + 1}/10], Loss: {average_loss:.4f}')
161
162 log("Final training completed.")
163 save_path = '/Users/...
164 torch.save(model.state_dict(), os.path.join(save_path, 'Model.pth'))
165 log("Best model saved.")
166
167 # Validation Loop
168 model.eval()
169 val_loss = 0.0
170
171 with torch.no_grad():
172     for batch in val_loader:
173         inputs, targets = batch
174         outputs = model(inputs)
175         loss = criterion(outputs, targets)
176         val_loss += loss.item()
177
178 average_val_loss = val_loss / len(val_loader)
179 log(f'Validation Loss: {average_val_loss:.4f}')
180
181 log(f"Validation completed.")
182
183 model.eval()
184 test_loss = 0.0
185
186 with torch.no_grad():
187     for batch in test_loader:
188         inputs, targets = batch
189         outputs = model(inputs)
190         loss = criterion(outputs, targets)
191         test_loss += loss.item()
192
193 average_test_loss = test_loss / len(test_loader)
194 log(f'Test Loss: {average_test_loss:.4f}')
195 log(f"Test completed.")
```

Listing A.4: Skrypt do optymalizacji sieci Neuronowej MLP z wykorzystaniem Nauczania Maszynowego.

```
1
2 import torch
```

```

3 import torch.nn as nn
4 import torch.nn.functional as F
5 import torch.optim as optim
6 import sqlite3
7 import pandas as pd
8 import numpy as np
9 from sklearn.model_selection import train_test_split
10 from torch.utils.data import DataLoader, TensorDataset
11 from datetime import datetime
12 import os
13 from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
14 from tqdm import tqdm
15 from torch.optim.lr_scheduler import ReduceLROnPlateau
16
17
18 def log(message):
19     timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
20     print(f"{timestamp} - {message}")
21
22
23 material_mapping = {'D1': 0, 'H1': 1, 'J1': 0, 'M3': 2, 'R1': 2, 'T1': 1}
24 num_material_types = len(set(material_mapping.values()))
25
26
27 class MaterialAwareNeuralNetwork(nn.Module):
28     def __init__(self, hidden_layer_size, num_material_types, activation_function,
29                 dropout_rate, negative_slope=0.01):
30         super(MaterialAwareNeuralNetwork, self).__init__()
31         self.fc1 = nn.Linear(2 + num_material_types, hidden_layer_size)
32         self.fc2 = nn.Linear(hidden_layer_size, hidden_layer_size)
33         self.fc3 = nn.Linear(hidden_layer_size, hidden_layer_size)
34         self.fc4 = nn.Linear(hidden_layer_size, 2)
35
36         self.activation_function = activation_function
37         self.negative_slope = negative_slope
38         self.dropout = nn.Dropout(p=dropout_rate)
39
40     def forward(self, x, material_type_one_hot):
41         x = torch.cat((x, material_type_one_hot), dim=1)
42         x = self._activate(self.fc1(x))
43         x = self.dropout(self._activate(self.fc2(x)))
44         x = self.dropout(self._activate(self.fc3(x)))
45         return self.fc4(x)
46
47     def _activate(self, layer):
48         if self.activation_function == "LeakyReLU":
49             return F.leaky_relu(layer, negative_slope=self.negative_slope)
50         elif self.activation_function == "ReLU":
51             return F.relu(layer)

```

```

51         else:
52             raise ValueError("Unsupported activation function")
53
54
55 def create_data_loaders(batch_size, database_path):
56     conn = sqlite3.connect(database_path)
57     query = "SELECT Material, SPT_File, SPT_Force, SPT_Distance, UTT_Force,
58             UTT_Distance FROM paired_data"
59     data = pd.read_sql_query(query, conn)
60     conn.close()
61
62     data['SPT_Force'] = pd.to_numeric(data['SPT_Force'], errors='coerce').fillna(0)
63     data['SPT_Distance'] = pd.to_numeric(data['SPT_Distance'], errors='coerce').fillna(0)
64     data['UTT_Force'] = pd.to_numeric(data['UTT_Force'], errors='coerce').fillna(0)
65     data['UTT_Distance'] = pd.to_numeric(data['UTT_Distance'], errors='coerce').fillna(0)
66
67     data['MaterialType'] = data['Material'].map(material_mapping)
68     data['CurveID'] = data[['Material', 'SPT_File']].astype(str).agg('__'.join, axis=1)
69
70     grouped = data.groupby('Material')['CurveID'].unique()
71     required_test_keys = []
72
73     for material, curve_ids in grouped.items():
74         if len(curve_ids) > 0:
75             selected = np.random.choice(curve_ids)
76             required_test_keys.append(selected)
77
78     remaining_keys = list(set(data['CurveID'].unique()) - set(required_test_keys))
79     remaining_train_keys, remaining_test_keys = train_test_split(remaining_keys,
80     test_size=0.2, random_state=42)
81     final_test_keys = list(set(remaining_test_keys).union(required_test_keys))
82     val_keys, test_keys = train_test_split(final_test_keys, test_size=0.5,
83     random_state=42)
84     train_keys = remaining_train_keys
85     is_test_sample = data['CurveID'].isin(test_keys)
86     test_ids_df = data.loc[is_test_sample, ['Material', 'SPT_File']].drop_duplicates()
87
88     train_data = data[data['CurveID'].isin(train_keys)]
89     val_data = data[data['CurveID'].isin(val_keys)]
90     test_data = data[data['CurveID'].isin(test_keys)]
91
92     def prepare_tensor(df):
93         features = df[['SPT_Force', 'SPT_Distance']].values
94         material_type = df['MaterialType'].values
95         material_one_hot = np.eye(num_material_types)[material_type]
96         features_combined = np.hstack([features, material_one_hot])

```

```

94     targets = df[['UTT_Force', 'UTT_Distance']].values
95     return torch.tensor(features_combined, dtype=torch.float32), torch.tensor(
96         targets, dtype=torch.float32)
97
98     train_features, train_targets = prepare_tensor(train_data)
99     val_features, val_targets = prepare_tensor(val_data)
100     test_features, test_targets = prepare_tensor(test_data)
101
102     train_dataset = TensorDataset(train_features, train_targets)
103     val_dataset = TensorDataset(val_features, val_targets)
104     test_dataset = TensorDataset(test_features, test_targets)
105
106     train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
107     val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=False)
108     test_loader = DataLoader(test_dataset, batch_size=batch_size, shuffle=False)
109
110     return train_loader, val_loader, test_loader, test_ids_df
111
112 # Parameters from Optuna
113 best_lr = 1.6802703407002962e-05
114 best_hidden_layer_size = 256
115 best_batch_size = 64
116 best_activation_function = "LeakyReLU"
117 best_dropout_rate = 0.24
118 best_negative_slope = 0.16
119
120 database_path = '/Users/.../material_data_MLP.db'
121 save_path = '/Users/...'
122
123 train_loader, val_loader, test_loader, test_ids_df = create_data_loaders(
124     best_batch_size, database_path)
125 model = MaterialAwareNeuralNetwork(best_hidden_layer_size, num_material_types,
126     best_activation_function,
127     best_dropout_rate, best_negative_slope)
128 optimizer = optim.Adam(model.parameters(), lr=best_lr)
129 scheduler = ReduceLROnPlateau(optimizer, mode='min', factor=0.5, patience=2, min_lr=1e
130     -7)
131 criterion = nn.MSELoss()
132
133 best_val_loss = float('inf')
134 patience = 20
135 patience_counter = 0
136 num_epochs = 50
137
138 log(f"Starting training")
139 for epoch in range(num_epochs):
140     model.train()
141     epoch_loss = 0.0

```

```

139
140     train_batches = tqdm(train_loader, desc=f"Epoch {epoch + 1}/{num_epochs}", leave=
False)
141     for batch in train_batches:
142         inputs, targets = batch
143         material_type_one_hot = F.one_hot(inputs[:, -num_material_types:].argmax(dim
=1),
144                                         num_classes=num_material_types).float()
145         optimizer.zero_grad()
146         outputs = model(inputs[:, :-num_material_types], material_type_one_hot)
147         loss = criterion(outputs, targets)
148         loss.backward()
149         optimizer.step()
150         epoch_loss += loss.item()
151         train_batches.set_postfix(loss=loss.item())
152
153     average_loss = epoch_loss / len(train_loader)
154
155     # Validation
156     model.eval()
157     val_loss = 0.0
158     with torch.no_grad():
159         for batch in val_loader:
160             inputs, targets = batch
161             material_type_one_hot = F.one_hot(inputs[:, -num_material_types:].argmax(
dim=1),
162                                             num_classes=num_material_types).float()
163             outputs = model(inputs[:, :-num_material_types], material_type_one_hot)
164             loss = criterion(outputs, targets)
165             val_loss += loss.item()
166     average_val_loss = val_loss / len(val_loader)
167     scheduler.step(average_val_loss)
168
169     if len(optimizer.param_groups) > 0:
170         current_lr = optimizer.param_groups[0]['lr']
171         log(f"Current LR: {current_lr:.8f}")
172
173     log(f"Epoch [{epoch + 1}/{num_epochs}], Train Loss: {average_loss:.5f}, Val Loss:
{average_val_loss:.5f}")
174
175     # Early stopping
176     if average_val_loss < best_val_loss:
177         best_val_loss = average_val_loss
178         patience_counter = 0
179         torch.save(model.state_dict(), os.path.join(save_path, 'ModelMLP.pth'))
180         log("Model saved.")
181     else:
182         patience_counter += 1
183         log(f"Early stopping counter: {patience_counter}/{patience}")

```

```

184         if patience_counter >= patience:
185             log("Early stopping triggered.")
186             break
187
188     log("Training DONE")
189
190     # Validation Loop
191     model.eval() # Set the model to evaluation mode
192     val_loss = 0.0
193
194     with torch.no_grad():
195         for batch in val_loader:
196             inputs, targets = batch
197             material_type_one_hot = F.one_hot(inputs[:, -num_material_types:].argmax(dim
198 =1),
199                                             num_classes=num_material_types).float()
200             outputs = model(inputs[:, :-num_material_types], material_type_one_hot)
201             loss = criterion(outputs, targets)
202             val_loss += loss.item()
203
204     average_val_loss = val_loss / len(val_loader)
205     log(f'Validation Loss: {average_val_loss:.4f}')
206     log(f"Validation DONE")
207
208     # Load best model
209     model.load_state_dict(torch.load(os.path.join(save_path, 'ModelMLP.pth')))
210     model.eval()
211
212     all_preds = []
213     all_targets = []
214
215     with torch.no_grad():
216         for batch in test_loader:
217             inputs, targets = batch
218             material_type_one_hot = F.one_hot(inputs[:, -num_material_types:].argmax(dim
219 =1),
220                                             num_classes=num_material_types).float()
221             outputs = model(inputs[:, :-num_material_types], material_type_one_hot)
222             all_preds.append(outputs.numpy())
223             all_targets.append(targets.numpy())
224
225     all_preds = np.vstack(all_preds)
226     all_targets = np.vstack(all_targets)
227
228     mse = mean_squared_error(all_targets, all_preds)
229     rmse = np.sqrt(mse)
230     mae = mean_absolute_error(all_targets, all_preds)
231     r2 = r2_score(all_targets, all_preds)

```

```

231 log(f"TEST RESULTS:")
232 log(f"Mean Absolute Error (MAE): {mae:.5f}")
233 log(f"Root Mean Squared Error (RMSE): {rmse:.5f}")
234 log(f"R2 Score: {r2:.5f}")
235 log(f"Test completed.")
236
237 # Save test results
238 results_df = pd.DataFrame({
239     'MAE': [mae],
240     'RMSE': [rmse],
241     'R2': [r2],
242     'Timestamp': [datetime.now().strftime('%Y-%m-%d %H:%M:%S')]
243 })
244
245 results_path = os.path.join(save_path, 'test_results_metrics.csv')
246 results_df.to_csv(results_path, index=False)
247 log(f"Saved test performance to: {results_path}")
248
249 test_ids_path = os.path.join(save_path, 'test_sample_ids.csv')
250 test_ids_df.to_csv(test_ids_path, index=False)
251 log(f"Saved test sample IDs to: {test_ids_path}")

```

Listing A.5: Skrypt do uczenia modelu w architekturze MLP na podstawie parametrów uzyskanych w procesie optymalizacji.

```

1
2 import os
3 import pandas as pd
4 import torch
5 import torch.nn as nn
6 import torch.nn.functional as F
7 import matplotlib.pyplot as plt
8 import numpy as np
9 import glob
10 from generate_re_rm_summary_v3 import analyze_and_plot_all
11
12 MATERIAL_CONSTANTS = {
13     '10H2M': {'S0': 19.46, 'L0': 19.93},
14     '13HMF': {'S0': 19.315, 'L0': 19.975},
15     '15HM': {'S0': 19.61, 'L0': 19.95},
16     'Other': {'S0': 19.462, 'L0': 19.952}
17 }
18
19 material_mapping = {'10H2M': 0, '13HMF': 2, '15HM': 1}
20 num_material_types = len(material_mapping)
21
22
23 class MaterialAwareNeuralNetwork(nn.Module):

```

```

24 def __init__(self, hidden_layer_size, num_material_types, activation_function,
25 dropout_rate, negative_slope=0.01):
26     super(MaterialAwareNeuralNetwork, self).__init__()
27     self.fc1 = nn.Linear(2 + num_material_types, hidden_layer_size)
28     self.fc2 = nn.Linear(hidden_layer_size, hidden_layer_size)
29     self.fc3 = nn.Linear(hidden_layer_size, hidden_layer_size)
30     self.fc4 = nn.Linear(hidden_layer_size, 2)
31
32     self.activation_function = activation_function
33     self.negative_slope = negative_slope
34     self.dropout = nn.Dropout(p=dropout_rate)
35
36 def forward(self, x, material_type_one_hot):
37     x = torch.cat((x, material_type_one_hot), dim=1)
38     x = self._activate(self.fc1(x))
39     x = self.dropout(self._activate(self.fc2(x)))
40     x = self.dropout(self._activate(self.fc3(x)))
41     return self.fc4(x)
42
43 def _activate(self, layer):
44     return F.leaky_relu(layer, negative_slope=self.negative_slope)
45
46 def normalize_data(file_path, output_folder):
47     df = pd.read_csv(file_path, sep='\t', decimal=',', usecols=['Force', 'Distance'])
48     df['Force'] = df['Force'].clip(lower=0)
49     df['Distance'] = df['Distance'].clip(lower=0)
50     normalized_force = df['Force'] / 1800.0
51     normalized_distance = df['Distance'] / 1.8
52     normalized_df = pd.DataFrame({
53         'Normalized Force': normalized_force,
54         'Normalized Distance': normalized_distance
55     })
56     if not os.path.exists(output_folder):
57         os.makedirs(output_folder)
58     output_file_path = os.path.join(output_folder, 'normalized_' + os.path.basename(
59         file_path))
60     normalized_df.to_csv(output_file_path, index=False, decimal=',', sep='\t')
61
62 def get_user_input():
63     print("Select material type:")
64     for i, material in enumerate(material_mapping.keys()):
65         print(f"{i + 1}. {material}")
66     print(f"{len(material_mapping) + 1}. Other")
67
68     choice = input("Choose material for analysis: ")
69     try:
70         choice = int(choice)

```



```

71     if 1 <= choice <= len(material_mapping):
72         selected_material = list(material_mapping.keys())[choice - 1]
73         return selected_material, material_mapping[selected_material]
74     else:
75         return "Other", -1
76 except ValueError:
77     print("Invalid input. Defaulting to 'Other'.")
78     return "Other", -1
79
80
81 def generate Utt_data(source_file, model, output_folder, material_index):
82     spt_data = pd.read_csv(source_file, sep='\t', decimal=',', usecols=['Normalized
83     Force', 'Normalized Distance'])
84     features = spt_data[['Normalized Force', 'Normalized Distance']].values
85     features_tensor = torch.tensor(features, dtype=torch.float32)
86
87     if material_index != -1:
88         material_type_one_hot = F.one_hot(torch.tensor([material_index]), num_classes=
89         num_material_types).float()
90     else:
91         material_type_one_hot = torch.ones((1, num_material_types)) /
92         num_material_types
93     material_type_one_hot = material_type_one_hot.repeat(features_tensor.shape[0], 1)
94
95     with torch.no_grad():
96         predictions = model(features_tensor, material_type_one_hot)
97
98     predicted_data = predictions.numpy()
99     predicted_data[:, 0] *= 13500.0
100    predicted_data[:, 1] *= 2.7
101    predicted_df = pd.DataFrame(predicted_data, columns=['UTT_Force', 'UTT_Distance'])
102    predicted_df['UTT_Distance'] = predicted_df['UTT_Distance'].clip(lower=0)
103    predicted_df.sort_values('UTT_Distance', inplace=True)
104    predicted_df.reset_index(drop=True, inplace=True)
105    if not os.path.exists(output_folder):
106        os.makedirs(output_folder)
107    file_name, _ = os.path.splitext(os.path.basename(source_file))
108    output_file_path = os.path.join(output_folder, file_name.replace('normalized_', ''
109    ) + '_UTT.csv')
110    predicted_df.to_csv(output_file_path, index=False)
111
112
113 def draw_graphs_and_calculate_properties(root_directory, selected_material):
114     files_pattern = os.path.join(root_directory, '**', '*.csv')
115
116     for file_path in glob.glob(files_pattern, recursive=True):
117         predicted_df = pd.read_csv(file_path)
118         predicted_df.dropna(inplace=True)
119         filename = os.path.basename(file_path)

```

```

116     h0_measured = int(filename.split('_')[0])
117     h0_ref = 500
118     correction_factor = h0_ref / h0_measured
119     S0 = MATERIAL_CONSTANTS[selected_material]['S0']
120     L0 = MATERIAL_CONSTANTS[selected_material]['L0']
121
122     predicted_df['Stress'] = (predicted_df['UTT_Force'] / S0) * correction_factor
123     predicted_df['Strain'] = predicted_df['UTT_Distance'] / L0
124     predicted_df.sort_values('Strain', inplace=True)
125     predicted_df.reset_index(drop=True, inplace=True)
126
127     if selected_material == '10H2M':
128         last_10_percent = int(len(predicted_df) * 0.20)
129         if last_10_percent >= 2:
130             tail_df = predicted_df.iloc[-last_10_percent:]
131             stress_diffs = tail_df['Stress'].diff().abs()
132             spike_indices = stress_diffs[stress_diffs > 50].index
133
134             if not spike_indices.empty:
135                 spike_positions = np.where(stress_diffs > 50)[0]
136                 if len(spike_positions) > 0:
137                     first_spike_pos = spike_positions[0]
138
139                     global_spike_idx = tail_df.index[first_spike_pos]
140                     if global_spike_idx > 0:
141                         flatten_value = predicted_df.loc[global_spike_idx - 1, '
142 Stress']
143
144                         predicted_df.loc[global_spike_idx:, 'Stress'] =
145                         flatten_value
146
147                         print(
148                             f"Flattened stress from index {global_spike_idx} onward
149                             due to spike > 50 MPa in last 20% of {filename}")
150
151     curves_directory = os.path.join(root_directory, 'Curves')
152     os.makedirs(curves_directory, exist_ok=True)
153     curve_file_name = os.path.basename(file_path).replace('.csv', '_stress_strain.
154 csv')
155     curve_file_path = os.path.join(curves_directory, curve_file_name)
156     predicted_df[['Strain', 'Stress']].to_csv(curve_file_path, index=False)
157
158     analyze_and_plot_all(
159         curve_folder=os.path.join(root_directory, 'Curves'),
160         material=selected_material
161     )
162
163 def process_all_steps(root_directory, model_path):

```

```

160     model = MaterialAwareNeuralNetwork(hidden_layer_size=256, num_material_types=
161                                         num_material_types,
162                                         activation_function="LeakyReLU", dropout_rate
163                                         =0.23818571240876843,
164                                         negative_slope=0.15914756121101373)
165
166     model.load_state_dict(torch.load(model_path))
167     model.eval()
168
169     selected_material, material_index = get_user_input()
170
171     normalized_root = os.path.join(root_directory, 'Normalized')
172     process_all_files(root_directory, normalized_root, normalize_data)
173
174     utt_root = os.path.join(root_directory, 'UTT')
175     process_all_files(normalized_root, utt_root, generate_utt_data, model,
176                     material_index)
177
178     draw_graphs_and_calculate_properties(utt_root, selected_material)
179
180 def process_all_files(source_root, output_root, process_function, model=None,
181                     material_index=None):
182     for dirpath, dirnames, filenames in os.walk(source_root):
183         for filename in filenames:
184             if filename.endswith('.txt'):
185                 file_path = os.path.join(dirpath, filename)
186                 relative_path = os.path.relpath(dirpath, source_root)
187                 output_folder = os.path.join(output_root, relative_path)
188                 if process_function == generate_utt_data:
189                     process_function(file_path, model, output_folder, material_index)
190                 else:
191                     process_function(file_path, output_folder)
192                 print(f'Processed file: {filename}')
193
194 # Paths
195 root_directory = (input("Enter the path to the root directory: ").strip().strip('"').
196                  strip("'"))
197 model_path = '/Users/.../ModelMLP.pth'
198
199 # Process all steps
200 process_all_steps(root_directory, model_path)

```

Listing A.6: Program do automatyzacji oceny stanu materiału dla modelu MLP

```

1
2 from __future__ import annotations
3
4 import warnings

```

```

5 from pathlib import Path
6 from typing import Dict, List, Tuple
7
8 import matplotlib.pyplot as plt
9 import numpy as np
10 import openpyxl
11 import pandas as pd
12 import scipy.stats as stats
13 from openpyxl.styles import PatternFill
14
15 BINS = 15
16 Z_THRESHOLD = 3.5
17 ALPHA = 0.05
18 NON_NORMAL_DISTs = ["expon", "lognorm", "gamma", "beta", "weibull_min"]
19 PLOT_DPI = 300
20 CI_LEVEL = 0.95
21 N_SIM = 2000
22
23 warnings.filterwarnings("ignore", category=RuntimeWarning)
24
25 def clean_series(series: pd.Series) -> Tuple[pd.Series, int]:
26     data = series.dropna()
27     if data.empty:
28         return data, 0
29     med = np.median(data)
30     mad = np.median(np.abs(data - med)) or 1e-6
31     z = 0.6745 * (data - med) / mad
32     mask = np.abs(z) <= Z_THRESHOLD
33     return data[mask].reset_index(drop=True), int((~mask).sum())
34
35 def best_fit_distribution(data: np.ndarray) -> Tuple[str, float, float]:
36     best_name, best_p, best_ks = "norm", 0.0, np.inf
37     for dname in NON_NORMAL_DISTs:
38         dist = getattr(stats, dname)
39         try:
40             params = dist.fit(data)
41             ks_stat, ks_p = stats.kstest(data, dname, args=params)
42             if ks_p > best_p:
43                 best_name, best_p, best_ks = dname, ks_p, ks_stat
44         except Exception:
45             continue
46     return best_name, best_p, best_ks
47
48 def plot_hist_with_fit(data, dist_name, mean, std, title, xlabel, legend, save_path,
49                        color):
50     plt.figure(figsize=(10, 7))
51     plt.hist(data, bins=BINS, density=True, alpha=0.6, color=color, edgecolor="black")
52     x_vals = np.linspace(min(data), max(data), 200)
53     if dist_name == "norm":

```

```

53     pdf_vals = stats.norm.pdf(x_vals, mean, std)
54     else:
55         dist = getattr(stats, dist_name)
56         pdf_vals = dist.pdf(x_vals, *dist.fit(data))
57     plt.plot(x_vals, pdf_vals, color=color, linewidth=2, label=legend)
58     median_val = np.median(data)
59     plt.axvline(mean, color=color, linestyle="dashed", linewidth=1.5, label=f"Mean = {
mean:.2f}")
60     plt.axvline(median_val, color=color, linestyle="dashdot", linewidth=1.5, label=f"
Median = {median_val:.2f}")
61     plt.title(title)
62     plt.xlabel(xlabel)
63     plt.ylabel("Density")
64     plt.legend()
65     plt.grid(True)
66     plt.tight_layout()
67     plt.savefig(save_path, dpi=PLOT_DPI)
68     plt.close()
69
70 def _prob_envelope(n, dist_name, params, ci):
71     alpha = 1.0 - ci
72     if dist_name == "norm":
73         loc, scale = params
74         sims = stats.norm.rvs(loc=loc, scale=scale, size=(N_SIM, n))
75     else:
76         dist = getattr(stats, dist_name)
77         sims = dist.rvs(*params, size=(N_SIM, n))
78     sims.sort(axis=1)
79     lower = np.percentile(sims, alpha / 2 * 100, axis=0)
80     upper = np.percentile(sims, (1 - alpha / 2) * 100, axis=0)
81     return lower, upper
82
83 def plot_probability(data, dist_name, title, save_path, ci_level=CI_LEVEL, sheet_label
=None):
84     plt.figure(figsize=(7, 7))
85     data_sorted = np.sort(data)
86     n = len(data_sorted)
87     probs = (np.arange(1, n + 1) - 0.5) / n
88     if dist_name == "norm":
89         params = (data_sorted.mean(), data_sorted.std(ddof=1))
90         dist_obj = stats.norm
91     else:
92         dist_obj = getattr(stats, dist_name)
93         params = dist_obj.fit(data_sorted)
94     theor = dist_obj.ppf(probs, *params)
95     plt.scatter(theor, data_sorted, s=15, label=f"Data vs {dist_name}")
96     slope, intercept = np.polyfit(theor, data_sorted, 1)
97     plt.plot(theor, slope * theor + intercept, color="red", linewidth=1.5, label=f"{
dist_name} fit")

```

```

98     lower, upper = _prob_envelope(n, dist_name, params, ci_level)
99     plt.fill_between(theor, lower, upper, color="grey", alpha=0.3, label=f"{int(
ci_level * 100)} % envelope")
100     y_unit = f"{sheet_label} [MPa]" if sheet_label else "[MPa]"
101     plt.xlabel("Theoretical Quantiles")
102     plt.ylabel(f"Ordered Values {y_unit}")
103     plt.title(title)
104     plt.legend()
105     plt.grid(True)
106     plt.tight_layout()
107     plt.savefig(save_path, dpi=PLOT_DPI)
108     plt.close()
109
110 def plot_overlay(sheet, sheet_name, out_dir):
111     prefixes = {
112         c.replace(" UTT", "").replace(" SPT", "")
113         for c in sheet.columns
114         if "UTT" in c or "SPT" in c
115     }
116     for prefix in prefixes:
117         utt_col, spt_col = f"{prefix} UTT", f"{prefix} SPT"
118         if utt_col not in sheet or spt_col not in sheet:
119             continue
120         utt, spt = sheet[utt_col].dropna(), sheet[spt_col].dropna()
121         if utt.empty or spt.empty:
122             continue
123         plt.figure(figsize=(10, 7))
124         plt.hist(utt, bins=BINS, alpha=0.5, density=True, edgecolor="black", color="
blue", label=f"{prefix} UTT")
125         plt.hist(spt, bins=BINS, alpha=0.5, density=True, edgecolor="black", color="
orange", label=f"{prefix} SPT")
126         x_vals = np.linspace(min(utt.min(), spt.min()), max(utt.max(), spt.max()),
200)
127         for data, color, lbl in [(utt, "blue", "UTT"), (spt, "orange", "SPT")]:
128             dist_name = "norm" if stats.shapiro(data).pvalue > ALPHA else
best_fit_distribution(data.values)[0]
129             try:
130                 if dist_name == "norm":
131                     pdf = stats.norm.pdf(x_vals, data.mean(), data.std(ddof=1))
132                 else:
133                     dist = getattr(stats, dist_name)
134                     pdf = dist.pdf(x_vals, *dist.fit(data))
135                 plt.plot(x_vals, pdf, color=color, linewidth=2, label=f"{lbl} {
dist_name} fit")
136             except Exception:
137                 continue
138         mean_utt, mean_spt = utt.mean(), spt.mean()
139         med_utt, med_spt = np.median(utt), np.median(spt)

```

```

140     plt.axvline(mean_utt, color="blue", linestyle="dashed", linewidth=1.5, label="
UTT mean")
141     plt.axvline(mean_spt, color="orange", linestyle="dashed", linewidth=1.5, label
="SPT mean")
142     plt.axvline(med_utt, color="blue", linestyle="dashdot", linewidth=1.5, label="
UTT median")
143     plt.axvline(med_spt, color="orange", linestyle="dashdot", linewidth=1.5, label
="SPT median")
144     delta_mean = abs(mean_utt - mean_spt)
145     delta_median = abs(med_utt - med_spt)
146     plt.title(f"Overlay Histogram - {sheet_name} - {prefix}\nDelta mean = {
delta_mean:.2f} MPa | Delta median = {delta_median:.2f} MPa")
147     plt.xlabel(f"{sheet_name} [MPa]")
148     plt.ylabel("Density")
149     plt.legend()
150     plt.grid(True)
151     plt.tight_layout()
152     plt.savefig(out_dir / f"{sheet_name}_{prefix}_UTT_vs_SPT_overlay.png", dpi=
PLOT_DPI)
153     plt.close()
154
155 def plot_combined_probability(sheet_df, sheet_name, out_dir, material):
156     markers = ["o", "s", "^", "D", "v", ">", "<", "p", "*", "x"]
157     colors = plt.cm.tab10.colors
158     plt.figure(figsize=(7, 7))
159     for i, col in enumerate(sheet_df.columns):
160         data = sheet_df[col].dropna().values
161         if data.size < 3:
162             continue
163         data_sorted = np.sort(data)
164         n = data_sorted.size
165         perc = (np.arange(1, n + 1) - 0.5) / n * 100
166         plt.scatter(data_sorted, perc, color=colors[i % len(colors)], marker=markers[i
% len(markers)], s=25, label=col)
167         mu, sigma = data_sorted.mean(), data_sorted.std(ddof=1)
168         z = np.linspace(stats.norm.ppf(0.001), stats.norm.ppf(0.999), 100)
169         plt.plot(mu + sigma * z, stats.norm.cdf(z) * 100, color=colors[i % len(colors)
], linestyle="--", linewidth=1)
170         plt.xlabel(f"{sheet_name} [MPa]")
171         plt.ylabel("Percent")
172         plt.title(f"Probability Plot of {material} {sheet_name}\nNormal - 95 % CI")
173         plt.grid(True)
174         plt.legend()
175         plt.tight_layout()
176         plt.savefig(out_dir / f"{sheet_name}_combined_probability.png", dpi=PLOT_DPI)
177         plt.close()
178
179 def apply_conditional_formatting(ws):
180     headers = [cell.value for cell in ws[1]]

```

```

181     if "Normality" not in headers:
182         return
183     idx = headers.index("Normality")
184     green = PatternFill(start_color="C6EFCE", end_color="C6EFCE", fill_type="solid")
185     red = PatternFill(start_color="FFC7CE", end_color="FFC7CE", fill_type="solid")
186     for row in ws.iter_rows(min_row=2):
187         cell = row[idx]
188         if cell.value == "Normal":
189             cell.fill = green
190         elif cell.value == "Not Normal":
191             cell.fill = red
192
193 def process_file(xlsx, out_base, sheets):
194     wb = openpyxl.load_workbook(xlsx)
195     for s in sheets:
196         if s in wb.sheetnames and f"{s}_raw" not in wb.sheetnames:
197             wb[s].title = f"{s}_raw"
198     wb.save(xlsx)
199     results = []
200     cleaned_sheets = {}
201     with pd.ExcelWriter(xlsx, mode="a", engine="openpyxl", if_sheet_exists="overlay")
202     as writer:
203         for s in sheets:
204             try:
205                 df_raw = pd.read_excel(xlsx, sheet_name=f"{s}_raw")
206             except Exception:
207                 continue
208             clean_df = pd.DataFrame()
209             for col in df_raw.columns:
210                 series, n_out = clean_series(df_raw[col])
211                 n_samples = series.size
212                 if series.empty:
213                     continue
214                 mean = series.mean()
215                 median = series.median()
216                 std_dev = series.std(ddof=1)
217                 ad_stat = stats.anderson(series, dist="norm").statistic
218                 shapiro_p = stats.shapiro(series).pvalue
219                 if shapiro_p > ALPHA:
220                     dist_name, ks_p, ks_stat = "norm", shapiro_p, np.nan
221                 else:
222                     dist_name, ks_p, ks_stat = best_fit_distribution(series.values)
223                 legend = (
224                     f"Normal Fit\nAD={ad_stat:.4f}, p={shapiro_p:.4f}"
225                     if dist_name == "norm"
226                     else f"{dist_name} Fit\nKS={ks_stat:.4f}, p={ks_p:.4f}"
227                 )
228                 results.append({
229                     "Sheet": s,

```



```

229         "Column": col,
230         "Mean": round(mean, 4),
231         "Median": round(median, 4),
232         "Std Dev": round(std_dev, 4),
233         "AD Statistic": round(ad_stat, 4),
234         "Shapiro p": round(shapiro_p, 4),
235         "Normality": "Normal" if dist_name == "norm" else "Not Normal",
236         "Best-Fit Dist": dist_name,
237         "N": n_samples,
238         "Outliers Removed": n_out,
239     })
240     clean_df[col] = series
241     mat_dir = out_base / xlsx.stem
242     mat_dir.mkdir(exist_ok=True)
243     plot_hist_with_fit(series.values, dist_name, mean, std_dev, f"
Histogram - {s} - {col}", f"{s} [MPa]", legend, mat_dir / f"{s}_{col}_histogram_fit
.png", "blue" if "UTT" in col else "orange" if "SPT" in col else "gray")
244     plot_probability(series.values, dist_name, f"Probability Plot - {s} -
{col}", mat_dir / f"{s}_{col}_probability_plot.png", sheet_label=s)
245     if not clean_df.empty:
246         clean_df.to_excel(writer, sheet_name=s, index=False)
247         cleaned_sheets[s] = clean_df
248     if results:
249         pd.DataFrame(results).to_excel(writer, sheet_name="Results", index=False)
250 wb = openpyxl.load_workbook(xlsx)
251 if "Results" in wb.sheetnames:
252     apply_conditional_formatting(wb["Results"])
253     wb.save(xlsx)
254 mat_dir = out_base / xlsx.stem
255 for sheet_name, df_clean in cleaned_sheets.items():
256     plot_overlay(df_clean, sheet_name, mat_dir)
257     plot_combined_probability(df_clean, sheet_name, mat_dir, xlsx.stem)
258
259 def main():
260     folder_input = input("Enter the path to the folder containing Excel files: ")
261     folder = Path(folder_input.strip().strip('"').strip('\''))
262     if not folder.is_dir():
263         print("Invalid folder path - exiting.")
264         return
265     output_base = folder / "output_analysis"
266     output_base.mkdir(exist_ok=True)
267     sheets = ["Re", "Rm"]
268     for xlsx_file in folder.glob("*.xlsx"):
269         if xlsx_file.name.startswith("~$"):
270             continue
271         print(f"\nProcessing: {xlsx_file.name}")
272         try:
273             process_file(xlsx_file, output_base, sheets)
274         except Exception as exc:

```

```

275         print(f"    Error while processing {xlsx_file.name}: {exc}")
276     print("Analysis complete. Check the 'output_analysis' folder.")
277
278 if __name__ == "__main__":
279     main()

```

Listing A.7: Program do automatyzacji analizy statystycznej danych SPT i UTT

```

1
2 from __future__ import annotations
3
4 import warnings
5 from pathlib import Path
6 from typing import Dict, List, Tuple
7
8 import matplotlib.pyplot as plt
9 import numpy as np
10 import openpyxl
11 import pandas as pd
12 import scipy.stats as stats
13 from openpyxl.styles import PatternFill
14
15
16 def csv_to_excel(csv_path: Path) -> Path:
17     df = pd.read_csv(csv_path, engine="python", decimal=".")
18     if df.shape[1] < 3:
19         raise ValueError(f"{csv_path.name} needs at least 3 columns (class, Re, Rm).")
20
21     xlsx_path = csv_path.with_suffix(".xlsx")
22     with pd.ExcelWriter(xlsx_path, engine="openpyxl", mode="w") as writer:
23         df.iloc[:, 1].to_frame(name="Yield Strength").to_excel(
24             writer, sheet_name="Re", index=False
25         )
26         df.iloc[:, 2].to_frame(name="Ultimate Strength").to_excel(
27             writer, sheet_name="Rm", index=False
28         )
29     return xlsx_path
30
31
32 BINS = 15
33 Z_THRESHOLD = 3.5
34 ALPHA = 0.05 # Shapiro decision level
35 NON_NORMAL_DISTS = ["lognorm", "weibull_min"]
36 PLOT_DPI = 300
37 CI_LEVEL = 0.95
38 N_SIM = 2000
39
40 warnings.filterwarnings("ignore", category=RuntimeWarning)
41

```

```
42
43 def clean_series(series: pd.Series) -> Tuple[pd.Series, int]:
44     series = pd.to_numeric(series, errors="coerce")
45     data = series.dropna()
46     if data.empty:
47         return data, 0
48     med = np.median(data)
49     mad = np.median(np.abs(data - med)) or 1e-6
50     z = 0.6745 * (data - med) / mad
51     mask = np.abs(z) <= Z_THRESHOLD
52     return data[mask].reset_index(drop=True), int((~mask).sum())
53
54
55 def best_fit_distribution(data: np.ndarray) -> Tuple[str, float, float]:
56     best_name, best_p, best_ks = "norm", 0.0, np.inf
57     for dname in NON_NORMAL_DISTs:
58         dist = getattr(stats, dname)
59         try:
60             params = dist.fit(data)
61             ks_stat, ks_p = stats.kstest(data, dname, args=params)
62             if ks_p > best_p:
63                 best_name, best_p, best_ks = dname, ks_p, ks_stat
64         except Exception:
65             continue
66     return best_name, best_p, best_ks
67
68
69 def plot_hist_with_fit(
70     data: np.ndarray,
71     dist_name: str,
72     mean: float,
73     std: float,
74     title: str,
75     xlabel: str,
76     legend: str,
77     save_path: Path,
78     color: str,
79 ) -> None:
80     plt.figure(figsize=(10, 7))
81
82     plt.hist(
83         data,
84         bins=BINS,
85         density=True,
86         alpha=0.6,
87         color=color,
88         edgecolor="black",
89     )
90
```

```

91     x_vals = np.linspace(min(data), max(data), 200)
92
93     if dist_name == "norm":
94         pdf_vals = stats.norm.pdf(x_vals, mean, std)
95     else:
96         dist = getattr(stats, dist_name)
97         pdf_vals = dist.pdf(x_vals, *dist.fit(data))
98     plt.plot(x_vals, pdf_vals, color=color, linewidth=2, label=legend)
99
100     median_val = np.median(data)
101     plt.axvline(mean, color=color, linestyle="dashed", linewidth=1.5,
102               label=f"Mean = {mean:.2f}")
103     plt.axvline(median_val, color=color, linestyle="dashdot", linewidth=1.5,
104               label=f"Median = {median_val:.2f}")
105
106     plt.title(title)
107     plt.xlabel(xlabel)
108     plt.ylabel("Density")
109     plt.legend()
110     plt.grid(True)
111     plt.tight_layout()
112     plt.savefig(save_path, dpi=PLOT_DPI)
113     plt.close()
114
115
116 def _prob_envelope(
117     n: int, dist_name: str, params: Tuple, ci: float
118 ) -> Tuple[np.ndarray, np.ndarray]:
119     alpha = 1.0 - ci
120     if dist_name == "norm":
121         loc, scale = params
122         sims = stats.norm.rvs(loc=loc, scale=scale, size=(N_SIM, n))
123     else:
124         dist = getattr(stats, dist_name)
125         sims = dist.rvs(*params, size=(N_SIM, n))
126     sims.sort(axis=1)
127     lower = np.percentile(sims, alpha / 2 * 100, axis=0)
128     upper = np.percentile(sims, (1 - alpha / 2) * 100, axis=0)
129     return lower, upper
130
131
132 def plot_probability(
133     data: np.ndarray,
134     dist_name: str,
135     title: str,
136     save_path: Path,
137     ci_level: float = CI_LEVEL,
138     sheet_label: str | None = None,
139 ) -> None:

```

```

140 plt.figure(figsize=(7, 7))
141 data_sorted = np.sort(data)
142 n = len(data_sorted)
143 probs = (np.arange(1, n + 1) - 0.5) / n
144
145 if dist_name == "norm":
146     params = (data_sorted.mean(), data_sorted.std(ddof=1))
147     dist_obj = stats.norm
148 else:
149     dist_obj = getattr(stats, dist_name)
150     params = dist_obj.fit(data_sorted)
151
152 theor = dist_obj.ppf(probs, *params)
153
154 plt.scatter(theor, data_sorted, s=15, label=f>Data vs {dist_name})
155
156 slope, intercept = np.polyfit(theor, data_sorted, 1)
157 plt.plot(theor, slope * theor + intercept, color="red", linewidth=1.5,
158         label=f"{dist_name} fit")
159
160 lower, upper = _prob_envelope(n, dist_name, params, ci_level)
161 plt.fill_between(theor, lower, upper, color="grey", alpha=0.3,
162                 label=f"{int(ci_level * 100)} % envelope")
163
164 y_unit = f"{sheet_label} [MPa]" if sheet_label else "[MPa]"
165 plt.xlabel("Theoretical Quantiles")
166 plt.ylabel(f"Ordered Values {y_unit}")
167
168 plt.title(title)
169 plt.legend()
170 plt.grid(True)
171 plt.tight_layout()
172 plt.savefig(save_path, dpi=PLOT_DPI)
173 plt.close()
174
175
176 def plot_overlay(sheet: pd.DataFrame, sheet_name: str, out_dir: Path) -> None:
177     prefixes = {
178         c.replace(" UTT", "").replace(" MLP", "").replace(" CNN", "")
179         for c in sheet.columns
180         if any(tag in c for tag in ("UTT", "MLP", "CNN"))
181     }
182
183     colours = {"UTT": "blue", "MLP": "orange", "CNN": "green"}
184
185     for prefix in prefixes:
186         cols = {tag: f"{prefix} {tag}" for tag in ("UTT", "MLP", "CNN") if f"{prefix}
187             {tag}" in sheet.columns}
188         if len(cols) < 3:

```

```
188         continue
189
190     plt.figure(figsize=(10, 7))
191     for tag in ("UTT", "MLP", "CNN"):
192         data = pd.to_numeric(sheet[cols[tag]], errors="coerce").dropna()
193         if data.empty:
194             continue
195
196         plt.hist(
197             data,
198             bins=BINS,
199             alpha=0.5,
200             density=True,
201             edgecolor="black",
202             color=colours[tag],
203             label=f"{prefix} {tag}",
204         )
205
206         x_vals = np.linspace(data.min(), data.max(), 200)
207         dist_name = (
208             "norm"
209             if stats.shapiro(data).pvalue > ALPHA
210             else best_fit_distribution(data.values)[0]
211         )
212
213         try:
214             pdf = (
215                 stats.norm.pdf(x_vals, data.mean(), data.std(ddof=1))
216                 if dist_name == "norm"
217                 else getattr(stats, dist_name).pdf(x_vals, *getattr(stats,
218 dist_name).fit(data))
219             )
220
221             plt.plot(
222                 x_vals,
223                 pdf,
224                 color=colours[tag],
225                 linewidth=2,
226                 label=f"{tag} {dist_name} fit",
227             )
228         except Exception:
229             pass
230
231         plt.axvline(
232             data.mean(),
233             color=colours[tag],
234             linestyle="dashed",
235             linewidth=1.5,
236             label=f"{tag} mean",
```

```

236         )
237         plt.axvline(
238             np.median(data),
239             color=colours[tag],
240             linestyle="dashdot",
241             linewidth=1.5,
242             label=f"{tag} median",
243         )
244
245     plt.title(f"Overlay Histogram - {sheet_name} - {prefix}")
246     plt.xlabel(f"{sheet_name} [MPa]")
247     plt.ylabel("Density")
248     plt.grid(True)
249     handles, labels = plt.gca().get_legend_handles_labels()
250     if handles:
251         plt.legend()
252     plt.tight_layout()
253     plt.savefig(
254         out_dir / f"{sheet_name}_{prefix}_UTT_MLP_CNN_overlay.png",
255         dpi=PLOT_DPI,
256     )
257     plt.close()
258
259     plot_combined_probability(sheet, sheet_name, prefix, out_dir)
260
261
262 def plot_combined_probability(
263     sheet_df: pd.DataFrame,
264     sheet_name: str,
265     prefix: str,
266     out_dir: Path,
267 ) -> None:
268     colours = {"UTT": "blue", "MLP": "orange", "CNN": "green"}
269     markers = {"UTT": "o", "MLP": "s", "CNN": "^"}
270     plt.figure(figsize=(7, 7))
271
272     for tag in ("UTT", "MLP", "CNN"):
273         col = f"{prefix} {tag}"
274         if col not in sheet_df.columns:
275             continue
276
277         data = pd.to_numeric(sheet_df[col], errors="coerce").dropna().values
278         if data.size < 3:
279             continue
280
281         data_sorted = np.sort(data)
282         n = data_sorted.size
283         percent = (np.arange(1, n + 1) - 0.5) / n * 100
284

```

```

285     plt.scatter(
286         data_sorted,
287         percent,
288         color=colours[tag],
289         marker=markers[tag],
290         s=25,
291         label=f"{tag}",
292     )
293
294     mu, sigma = data_sorted.mean(), data_sorted.std(ddof=1)
295     x_vals = np.linspace(data_sorted.min(), data_sorted.max(), 200)
296     plt.plot(
297         x_vals,
298         stats.norm.cdf(x_vals, mu, sigma) * 100,
299         color=colours[tag],
300         linestyle="--",
301         linewidth=1,
302     )
303
304     plt.xlabel(f"{sheet_name} [MPa]")
305     plt.ylabel("Percent")
306     plt.title(f"Probability Plot of {prefix} {sheet_name}\nNormal - 95 % CI")
307     plt.grid(True)
308     plt.legend()
309     plt.tight_layout()
310     plt.savefig(
311         out_dir / f"{sheet_name}_{prefix}_combined_probability.png",
312         dpi=PLOT_DPI,
313     )
314     plt.close()
315
316
317 def apply_conditional_formatting(
318     ws: openpyxl.worksheet.worksheet.Worksheet,
319 ) -> None:
320     headers = [cell.value for cell in ws[1]]
321     if "Normality" not in headers:
322         return
323     idx = headers.index("Normality")
324
325     green = PatternFill(
326         start_color="C6EFCE", end_color="C6EFCE", fill_type="solid"
327     )
328     red = PatternFill(
329         start_color="FFC7CE", end_color="FFC7CE", fill_type="solid"
330     )
331
332     for row in ws.iter_rows(min_row=2):
333         cell = row[idx]

```



```

334         if cell.value == "Normal":
335             cell.fill = green
336         elif cell.value == "Not Normal":
337             cell.fill = red
338
339
340 def process_file(xlsx: Path, out_base: Path, sheets: List[str]) -> None:
341     wb = openpyxl.load_workbook(xlsx)
342     for s in sheets:
343         if s in wb.sheetnames and f"{s}_raw" not in wb.sheetnames:
344             wb[s].title = f"{s}_raw"
345     wb.save(xlsx)
346
347     results: List[Dict] = []
348     cleaned_sheets: Dict[str, pd.DataFrame] = {}
349
350     with pd.ExcelWriter(
351         xlsx, mode="a", engine="openpyxl", if_sheet_exists="overlay"
352     ) as writer:
353         for s in sheets:
354             try:
355                 df_raw = pd.read_excel(xlsx, sheet_name=f"{s}_raw")
356             except Exception:
357                 continue
358
359             clean_df = pd.DataFrame()
360             for col in df_raw.columns:
361                 series, n_out = clean_series(df_raw[col])
362                 series = pd.to_numeric(series, errors="coerce").dropna()
363                 n_samples = series.size
364                 if series.empty:
365                     continue
366
367                 mean = series.mean()
368                 median = series.median()
369                 std_dev = series.std(ddof=1)
370                 ad_stat = stats.anderson(series, dist="norm").statistic
371                 shapiro_p = stats.shapiro(series).pvalue
372
373                 if shapiro_p > ALPHA:
374                     dist_name, ks_p, ks_stat = "norm", shapiro_p, np.nan
375                 else:
376                     dist_name, ks_p, ks_stat = best_fit_distribution(
377                         series.values
378                     )
379
380                 legend = (
381                     f"Normal Fit\nAD={ad_stat:.4f}, p={shapiro_p:.4f}"
382                     if dist_name == "norm"

```

```

383         else f"{dist_name} Fit\nKS={ks_stat:.4f}, p={ks_p:.4f}"
384     )
385
386     results.append(
387         {
388             "Sheet": s,
389             "Column": col,
390             "Mean": round(mean, 0),
391             "Median": round(median, 0),
392             "Std Dev": round(std_dev, 2),
393             "AD Statistic": round(ad_stat, 3),
394             "Shapiro p": round(shapiro_p, 3),
395             "Normality": "Normal"
396             if dist_name == "norm"
397             else "Not Normal",
398             "Best-Fit Dist": dist_name,
399             "N": n_samples,
400             "Outliers Removed": n_out,
401         }
402     )
403
404     clean_df[col] = series
405
406     mat_dir = out_base / xlsx.stem
407     mat_dir.mkdir(exist_ok=True)
408     plot_hist_with_fit(
409         series.values,
410         dist_name,
411         mean,
412         std_dev,
413         f"Histogram - {s} - {col}",
414         f"{s} [MPa]",
415         legend,
416         mat_dir / f"{s}_{col}_histogram_fit.png",
417         "blue" if "UTT" in col else
418         "orange" if "MLP" in col else
419         "green" if "CNN" in col else
420         "gray",
421     )
422     plot_probability(
423         series.values,
424         dist_name,
425         f"Probability Plot - {s} - {col}",
426         mat_dir / f"{s}_{col}_probability_plot.png",
427         sheet_label=s,
428     )
429
430     if not clean_df.empty:
431         clean_df.to_excel(writer, sheet_name=s, index=False)

```

```

432         cleaned_sheets[s] = clean_df
433
434     if results:
435         pd.DataFrame(results).to_excel(
436             writer, sheet_name="Results", index=False
437         )
438
439     wb = openpyxl.load_workbook(xlsx)
440     if "Results" in wb.sheetnames:
441         apply_conditional_formatting(wb["Results"])
442         wb.save(xlsx)
443
444     mat_dir = out_base / xlsx.stem
445     for sheet_name, df_clean in cleaned_sheets.items():
446         plot_overlay(df_clean, sheet_name, mat_dir)
447
448
449 def main() -> None:
450     folder = Path(
451         input("Enter the path to the folder containing CSV or Excel files: ")
452         .strip().strip('"').strip('\'')
453     )
454     if not folder.is_dir():
455         print("Invalid folder path - exiting.")
456         return
457
458     output_base = folder / "output_analysis"
459     output_base.mkdir(exist_ok=True)
460
461     sheets = ["Re", "Rm"]
462
463     for f in folder.iterdir():
464         if f.name.startswith("~$"):
465             continue
466         try:
467             if f.suffix.lower() == ".csv":
468                 f = csv_to_excel(f)
469             elif f.suffix.lower() != ".xlsx":
470                 continue
471
472             print(f"\nProcessing: {f.name}")
473             process_file(f, output_base, sheets)
474         except Exception as exc:
475             print(f"Error {f.name}: {exc}")
476
477     print("Analysis DONE")
478
479
480 if __name__ == "__main__":

```

```
481 main()
```

Listing A.8: Program do automatyzacji analizy statystycznej UTT vs CNN vs MLP

```

1
2 import pandas as pd
3 import os
4 import json
5 import sqlite3
6 from datetime import datetime
7
8 def log(message):
9     timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
10    print(f"{timestamp} - {message}")
11
12 def convert_and_clean_data(df):
13     for col in ['Normalized Force', 'Normalized Distance']:
14         df[col] = df[col].astype(str).str.replace(',', '.').astype('float32')
15     return df
16
17 def load_strength_table(excel_path):
18     strength_df = pd.read_excel(excel_path, usecols=[1, 2, 3], header=0)
19     strength_df.columns = ['UTT_File', 'Re', 'Rm']
20     strength_df['UTT_File'] = strength_df['UTT_File'].astype(str).str.strip()
21     return strength_df.set_index('UTT_File').to_dict(orient='index')
22
23 def safe_read_csv(file_path, sep='\t', encoding='utf-8'):
24     try:
25         return pd.read_csv(file_path, sep=sep, encoding=encoding)
26     except UnicodeDecodeError:
27         return pd.read_csv(file_path, sep=sep, encoding='ascii')
28
29 def load_and_clean_curve(file_path):
30     df = safe_read_csv(file_path)
31     return convert_and_clean_data(df)[['Normalized Force', 'Normalized Distance']].values.tolist()
32
33 def create_samplewise_database(spt_root, utt_root, strength_excel_path, materials,
34                               output_path):
35     strength_map = load_strength_table(strength_excel_path)
36     samples = []
37
38     for material in materials:
39         spt_folder = os.path.join(spt_root, f"{material} SPT")
40         utt_folder = os.path.join(utt_root, f"{material} UTT")
41         spt_files = [f for f in os.listdir(spt_folder) if f.endswith('.txt')]
42         utt_files = [f for f in os.listdir(utt_folder) if f.endswith('.txt')]
43
44         for spt_file in spt_files:

```

```

44     spt_curve_path = os.path.join(spt_folder, spt_file)
45     spt_curve = load_and_clean_curve(spt_curve_path)
46
47     for utt_file in utt_files:
48         utt_curve_path = os.path.join(utt_folder, utt_file)
49         utt_curve = load_and_clean_curve(utt_curve_path)
50
51         utt_key = utt_file.strip()
52         if utt_key not in strength_map:
53             continue
54
55         Re = strength_map[utt_key]['Re']
56         Rm = strength_map[utt_key]['Rm']
57
58         samples.append({
59             'Material': material,
60             'SPT_File': spt_file,
61             'UTT_File': utt_file,
62             'SPT_Curve': json.dumps(spt_curve),
63             'UTT_Curve': json.dumps(utt_curve),
64             'Re': Re,
65             'Rm': Rm
66         })
67
68     if output_path.endswith(".db"):
69         conn = sqlite3.connect(output_path)
70         cur = conn.cursor()
71         cur.execute('DROP TABLE IF EXISTS paired_data')
72         cur.execute('''
73             CREATE TABLE IF NOT EXISTS paired_data (
74                 ID INTEGER PRIMARY KEY,
75                 Material TEXT,
76                 SPT_File TEXT,
77                 UTT_File TEXT,
78                 SPT_Curve TEXT,
79                 UTT_Curve TEXT,
80                 Re REAL,
81                 Rm REAL
82             )
83         ''')
84         for sample in samples:
85             cur.execute('''
86                 INSERT INTO paired_data (Material, SPT_File, UTT_File, SPT_Curve,
87                 UTT_Curve, Re, Rm)
88                 VALUES (?, ?, ?, ?, ?, ?, ?)
89             ''', (
90                 sample['Material'], sample['SPT_File'], sample['UTT_File'],
91                 sample['SPT_Curve'], sample['UTT_Curve'], sample['Re'], sample['Rm']

```

```

92     conn.commit()
93     conn.close()
94     log(f"Database saved as SQLite at: {output_path}")
95     else:
96         df = pd.DataFrame(samples)
97         df.to_csv(output_path, index=False)
98         log(f"Database saved as CSV at: {output_path}")
99
100     log(f"Total sample pairs saved: {len(samples)}")
101
102 spt_root = '/Users/.../SPT'
103 utt_root = '/Users/.../UTT'
104 materials = ['D1', 'H1', 'J1', 'M3', 'R1', 'T1']
105 output_path = "/Users/.../re_rm_database_v2.3.db"
106
107 create_samplewise_database(spt_root, utt_root, strength_excel_path, materials,
    output_path)

```

Listing A.9: Program do przygotowania bazy danych *Curve to Curve* z parami SPT i UTT danych do wykresów dla danego materiału.

```

1
2 import sqlite3
3 import json
4 import numpy as np
5 from tqdm import tqdm
6
7 def adaptive_sample_curve(curve, num_points=300):
8     curve = np.array(curve)
9     x, y = curve[:, 0], curve[:, 1]
10
11     if len(np.unique(x)) < 5 or np.allclose(x, x[0]):
12         idxs = np.linspace(0, len(curve) - 1, num_points).astype(int)
13         return curve[idxs].tolist()
14
15     try:
16         with np.errstate(divide='raise', invalid='raise'):
17             dy = np.gradient(y, x)
18             ddy = np.gradient(dy, x)
19             if (np.any(np.isnan(dy)) or np.any(np.isnan(ddy)) or
20                 np.any(np.isinf(dy)) or np.any(np.isinf(ddy))):
21                 raise FloatingPointError
22             weights = np.abs(dy) + 0.5 * np.abs(ddy)
23             weights = weights / np.sum(weights)
24             cdf = np.cumsum(weights)
25             cdf = cdf / cdf[-1]
26             uniform_samples = np.linspace(0, 1, num_points)
27             indices = np.searchsorted(cdf, uniform_samples)
28             indices = np.clip(indices, 0, len(curve) - 1)

```

```

29         return curve[indices].tolist()
30     except Exception:
31         idxs = np.linspace(0, len(curve) - 1, num_points).astype(int)
32         return curve[idxs].tolist()
33
34 def resample_database(input_db, output_db, num_points=300):
35     adaptive_count = 0
36     fallback_count = 0
37     src_conn = sqlite3.connect(input_db)
38     src_cursor = src_conn.cursor()
39     src_cursor.execute("SELECT * FROM paired_data")
40     rows = src_cursor.fetchall()
41     col_names = [desc[0] for desc in src_cursor.description]
42
43     dst_conn = sqlite3.connect(output_db)
44     dst_cursor = dst_conn.cursor()
45     dst_cursor.execute("DROP TABLE IF EXISTS paired_data")
46     dst_cursor.execute("""
47         CREATE TABLE paired_data (
48             ID INTEGER PRIMARY KEY,
49             Material TEXT,
50             SPT_File TEXT,
51             UTT_File TEXT,
52             SPT_Curve TEXT,
53             UTT_Curve TEXT,
54             Re REAL,
55             Rm REAL
56         )
57     """)
58
59     print("Resampling curves...")
60     for row in tqdm(rows):
61         row_dict = dict(zip(col_names, row))
62         spt = json.loads(row_dict['SPT_Curve'])
63         utt = json.loads(row_dict['UTT_Curve'])
64         try:
65             spt_small = adaptive_sample_curve(spt, num_points=num_points)
66             utt_small = adaptive_sample_curve(utt, num_points=num_points)
67             adaptive_count += 1
68         except:
69             spt_small = adaptive_sample_curve(spt, num_points=num_points)
70             utt_small = adaptive_sample_curve(utt, num_points=num_points)
71             fallback_count += 1
72
73         dst_cursor.execute("""
74             INSERT INTO paired_data (Material, SPT_File, UTT_File, SPT_Curve,
75             UTT_Curve, Re, Rm)
76             VALUES (?, ?, ?, ?, ?, ?, ?)
77         """, (

```

```

77         row_dict['Material'],
78         row_dict['SPT_File'],
79         row_dict['UTT_File'],
80         json.dumps(spt_small),
81         json.dumps(utt_small),
82         row_dict['Re'],
83         row_dict['Rm']
84     ))
85
86     dst_conn.commit()
87     src_conn.close()
88     dst_conn.close()
89     print(f"\nResampled DB saved to: {output_db}")
90     print(f"\nSummary:\nAdaptive sampling used: {adaptive_count}\nFallback (uniform)
91     sampling used: {fallback_count}\n")
92
93     with open("resample_summary.txt", "w") as f:
94         f.write(f"Adaptive: {adaptive_count}\n")
95         f.write(f"Fallback: {fallback_count}\n")
96
97 if __name__ == '__main__':
98     input_db_path = "/Users/.../re_rm_database_v2.3.db"
99     output_db_path = "/Users/.../re_rm_database_v2.3_small.db"
100     resample_database(input_db_path, output_db_path, num_points=300)

```

Listing A.10: Program do przygotowania zeskalowanej bazy danych *Curve to Curve*

```

1
2 import torch
3 import torch.nn as nn
4 import torch.nn.functional as F
5 from torch.utils.data import DataLoader, Dataset
6 import numpy as np
7 from tqdm import tqdm
8 import os
9 import pandas as pd
10 import matplotlib.pyplot as plt
11 import sqlite3
12 import json
13 import random
14 import time
15 from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
16 import optuna
17 import hashlib
18 from datetime import datetime
19
20 material_mapping = {'D1': 0, 'H1': 1, 'J1': 0, 'M3': 2, 'R1': 2, 'T1': 1}
21 num_material_classes = len(set(material_mapping.values()))
22 db_path = '/Users/.../re_rm_database_v2.3_small.db'

```



```

23 OUTPUT_DIR = '/Users/...'
24 os.makedirs(OUTPUT_DIR, exist_ok=True)
25 device = torch.device('cuda' if torch.cuda.is_available() else 'mps' if torch.backends
    .mps.is_available() else 'cpu')
26 TOTAL_TRIALS = 18
27
28 def tqdm_optuna_callback(study, trial):
29     if not hasattr(study, "_tqdm"):
30         study._tqdm = tqdm(total=TOTAL_TRIALS, desc="Optuna Trials")
31         study._start_time = datetime.now()
32     study._tqdm.update(1)
33     elapsed = (datetime.now() - study._start_time).total_seconds()
34     completed = study._tqdm.n
35     remaining = TOTAL_TRIALS - completed
36     avg_per_trial = elapsed / completed if completed > 0 else 0
37     eta = avg_per_trial * remaining
38     study._tqdm.set_postfix({"best": f"{study.best_value:.5f}", "ETA": f"{int(eta //
    60):02d}:{int(eta % 60):02d} min"})
39
40 class SPTtoUTTDataset(Dataset):
41     def __init__(self, db_path, split='train', test_split=0.2, val_split=0.1, seed=42)
    :
42         self.conn = sqlite3.connect(db_path)
43         self.split = split
44         self.seed = seed
45         df = pd.read_sql_query("SELECT * FROM paired_data", self.conn)
46         df['SPT_Curve'] = df['SPT_Curve'].apply(json.loads)
47         df['UTT_Curve'] = df['UTT_Curve'].apply(json.loads)
48         random.seed(seed)
49         self.train_set, self.val_set, self.test_set = self._split_data(df, test_split,
    val_split)
50         self.data = {'train': self.train_set, 'val': self.val_set, 'test': self.
    test_set}[split]
51
52     def _hash_curve(self, curve):
53         return hashlib.md5(np.array(curve).astype(np.float32).tobytes()).hexdigest()
54
55     def _split_data(self, df, test_split, val_split):
56         train_rows, val_rows, test_rows = [], [], []
57         grouped = df.groupby("Material")
58         for material, group in grouped:
59             group = group.copy()
60             group['SPT_ID'] = group['SPT_Curve'].apply(self._hash_curve)
61             group['UTT_ID'] = group['UTT_Curve'].apply(self._hash_curve)
62             unique_spt = group['SPT_ID'].unique().tolist()
63             unique_utt = group['UTT_ID'].unique().tolist()
64             random.shuffle(unique_spt)
65             random.shuffle(unique_utt)
66             n_spt, n_utt = len(unique_spt), len(unique_utt)

```

```

67     n_spt_test, n_spt_val = int(n_spt * test_split), int(n_spt * val_split)
68     n_utt_test, n_utt_val = int(n_utt * test_split), int(n_utt * val_split)
69     spt_test_ids = set(unique_spt[:n_spt_test])
70     spt_val_ids = set(unique_spt[n_spt_test:n_spt_test + n_spt_val])
71     spt_train_ids = set(unique_spt[n_spt_test + n_spt_val:])
72     utt_test_ids = set(unique_utt[:n_utt_test])
73     utt_val_ids = set(unique_utt[n_utt_test:n_utt_test + n_utt_val])
74     utt_train_ids = set(unique_utt[n_utt_test + n_utt_val:])
75     for _, row in group.iterrows():
76         spt_id, utt_id = row['SPT_ID'], row['UTT_ID']
77         if spt_id in spt_train_ids and utt_id in utt_train_ids:
78             train_rows.append(row)
79         elif spt_id in spt_val_ids and utt_id in utt_val_ids:
80             val_rows.append(row)
81         elif spt_id in spt_test_ids and utt_id in utt_test_ids:
82             test_rows.append(row)
83     train_df = pd.DataFrame(train_rows).reset_index(drop=True)
84     val_df = pd.DataFrame(val_rows).reset_index(drop=True)
85     test_df = pd.DataFrame(test_rows).reset_index(drop=True)
86     test_file_df = test_df[['SPT_File', 'Material']].copy()
87     test_file_df.to_csv(os.path.join(OUTPUT_DIR, "test_curve_ids.csv"), index=
False)
88     return train_df, val_df, test_df
89
90     def __len__(self):
91         return len(self.data)
92
93     def __getitem__(self, idx):
94         row = self.data.iloc[idx]
95         spt_curve = torch.tensor(np.array(row['SPT_Curve']).T, dtype=torch.float32)
96         utt_curve = torch.tensor(np.array(row['UTT_Curve']).T, dtype=torch.float32)
97         mat_vector = torch.tensor([1.0 if i == material_mapping[row['Material']] else
0.0 for i in range(num_material_classes)], dtype=torch.float32)
98         return spt_curve, mat_vector, utt_curve
99
100 class CurveToCurveCNN(nn.Module):
101     def __init__(self, hidden_size=512, dropout_rate=0.4, activation='relu'):
102         super().__init__()
103         self.activation = activation
104         self.dropout = nn.Dropout(dropout_rate)
105         self.conv1 = nn.Conv1d(5, hidden_size, kernel_size=7, padding=3)
106         self.conv2 = nn.Conv1d(hidden_size, hidden_size, kernel_size=7, padding=3)
107         self.conv3 = nn.Conv1d(hidden_size, hidden_size * 2, kernel_size=7, padding=3)
108         self.conv4 = nn.Conv1d(hidden_size * 2, hidden_size * 2, kernel_size=7,
padding=3)
109         self.conv5 = nn.Conv1d(hidden_size * 2, hidden_size, kernel_size=7, padding=3)
110         self.conv6 = nn.Conv1d(hidden_size, hidden_size, kernel_size=7, padding=3)
111         self.head_10H2M = nn.Conv1d(hidden_size, 2, kernel_size=1)
112         self.head_13HMF = nn.Conv1d(hidden_size, 2, kernel_size=1)

```

```

113         self.head_15HM = nn.Conv1d(hidden_size, 2, kernel_size=1)
114
115     def forward(self, x, mat_onehot):
116         B, C, L = x.shape
117         mat = mat_onehot.unsqueeze(2).expand(B, -1, L)
118         x = torch.cat([x, mat], dim=1)
119         x = self._act(self.conv1(x))
120         x = self._act(self.conv2(x))
121         x = self._act(self.conv3(x))
122         x = self.dropout(self._act(self.conv4(x)))
123         x = self._act(self.conv5(x))
124         x = self._act(self.conv6(x))
125         mat_indices = torch.argmax(mat_onehot, dim=1)
126         out = torch.zeros(B, 2, L, device=x.device)
127         for i in range(B):
128             if mat_indices[i] == 0:
129                 out[i] = self.head_10H2M(x[i].unsqueeze(0)).squeeze(0)
130             elif mat_indices[i] == 1:
131                 out[i] = self.head_15HM(x[i].unsqueeze(0)).squeeze(0)
132             elif mat_indices[i] == 2:
133                 out[i] = self.head_13HMF(x[i].unsqueeze(0)).squeeze(0)
134         return out
135
136     def _act(self, layer):
137         return F.leaky_relu(layer) if self.activation == 'leaky_relu' else F.relu(
138             layer)
139
140 def objective(trial):
141     hidden_size = trial.suggest_categorical("hidden_size", [256, 384, 512])
142     dropout_rate = trial.suggest_categorical("dropout", [0.4, 0.5, 0.6])
143     batch_size = trial.suggest_categorical("batch_size", [8, 16])
144     activation = 'relu'
145     model = CurveToCurveCNN(hidden_size=hidden_size, dropout_rate=dropout_rate,
146                             activation=activation).to(device)
147     train_loader = DataLoader(SPTtoUTTDataset(db_path, split='train'), batch_size=
148         batch_size, shuffle=True)
149     val_loader = DataLoader(SPTtoUTTDataset(db_path, split='val'), batch_size=
150         batch_size, shuffle=False)
151     optimizer = torch.optim.AdamW(model.parameters(), lr=1e-3)
152     criterion = nn.MSELoss()
153     best_val_loss = float('inf')
154     patience, counter = 10, 0
155     val_losses = []
156     start = time.time()
157     for epoch in tqdm(range(50), desc=f"Trial {trial.number:02d} - Epochs", leave=
158         False):
159         model.train()
160         for spt, mat, utt in train_loader:
161             spt, mat, utt = spt.to(device), mat.to(device), utt.to(device)

```

```

157         optimizer.zero_grad()
158         pred = model(spt, mat)
159         loss = criterion(pred, utt)
160         loss.backward()
161         optimizer.step()
162     model.eval()
163     val_loss = 0
164     with torch.no_grad():
165         for spt, mat, utt in val_loader:
166             spt, mat, utt = spt.to(device), mat.to(device), utt.to(device)
167             pred = model(spt, mat)
168             val_loss += criterion(pred, utt).item()
169     val_loss /= len(val_loader)
170     val_losses.append(val_loss)
171     if val_loss < best_val_loss:
172         best_val_loss = val_loss
173         torch.save(model.state_dict(), os.path.join(OUTPUT_DIR, f"
best_model_trial_{trial.number}.pt"))
174         counter = 0
175     else:
176         counter += 1
177         if counter >= patience:
178             break
179     if trial.number == 0:
180         pd.DataFrame({"epoch": list(range(1, len(val_losses) + 1)), "val_loss":
val_losses}).to_csv(
181             os.path.join(OUTPUT_DIR, f"learning_curve_trial_{trial.number}.csv"),
index=False)
182     duration = time.time() - start
183     with open(os.path.join(OUTPUT_DIR, "optuna_trial_log.csv"), "a") as f:
184         f.write(f"{trial.number},{hidden_size},{dropout_rate:.3f},{1e-3:.6f},{
batch_size},{activation},AdamW,{best_val_loss:.6f},{duration:.2f}\n")
185     return best_val_loss
186
187 if __name__ == '__main__':
188     with open(os.path.join(OUTPUT_DIR, "optuna_trial_log.csv"), "w") as f:
189         f.write("trial,hidden_size,dropout,lr,batch_size,activation,optimizer,val_loss
,time_sec\n")
190     study = optuna.create_study(direction="minimize")
191     study.optimize(objective, n_trials=TOTAL_TRIALS, callbacks=[tqdm_optuna_callback])
192     best_model_path = os.path.join(OUTPUT_DIR, f"best_model_trial_{study.best_trial.
number}.pt")
193     model = CurveToCurveCNN(
194         hidden_size=study.best_trial.params['hidden_size'],
195         dropout_rate=study.best_trial.params['dropout'],
196         activation=study.best_trial.params['activation']
197     ).to(device)
198     model.load_state_dict(torch.load(best_model_path))
199     model.eval()

```

```

200     test_loader = DataLoader(SPTtoUTTDataset(db_path, split='test'), batch_size=1,
201                             shuffle=False)
202     results = []
203     with torch.no_grad():
204         for i, (spt, mat, utt) in enumerate(test_loader):
205             spt, mat, utt = spt.to(device), mat.to(device), utt.to(device)
206             pred = model(spt, mat)
207             true = utt.cpu().squeeze().numpy().T
208             pred = pred.cpu().squeeze().numpy().T
209             results.append({
210                 "SampleID": i,
211                 "MAE": mean_absolute_error(true, pred),
212                 "MSE": mean_squared_error(true, pred),
213                 "R2": r2_score(true, pred)
214             })
215     df = pd.DataFrame(results)
216     df.to_csv(os.path.join(OUTPUT_DIR, "test_set_results_best_model Optuna v2.csv"),
217              index=False)

```

Listing A.11: Program do optymalizacji modelu sieci neuronowej CNN

```

1
2 import os
3 import json
4 import torch
5 import torch.nn as nn
6 import torch.nn.functional as F
7 from torch.utils.data import DataLoader
8 import numpy as np
9 import pandas as pd
10 from tqdm import tqdm
11 from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
12 import matplotlib.pyplot as plt
13 import sqlite3
14 import random
15 import hashlib
16 from time import time
17
18 # CONFIGURATION
19 TRIAL_ID = 1
20 OUTPUT_DIR = "..."
21 DB_PATH = "..."
22 BEST_MODEL_PATH = f"{OUTPUT_DIR}/best_model_pretrained.pt"
23 SAVE_MODEL_PATH = f"{OUTPUT_DIR}/final_model_trial_{TRIAL_ID}_retrained.pt"
24 device = torch.device("cuda" if torch.cuda.is_available() else "mps" if torch.backends
    .mps.is_available() else "cpu")
25
26 params = {
27     "hidden_size": 512,

```

```

28     "dropout": 0.4,
29     "batch_size": 16,
30     "activation": "relu",
31     "optimizer": "AdamW"
32 }
33
34 with open(f"{OUTPUT_DIR}/trial_{TRIAL_ID}_params.json", "w") as f:
35     json.dump(params, f, indent=2)
36
37 material_mapping = {'D1': 0, 'H1': 1, 'J1': 0, 'M3': 2, 'R1': 2, 'T1': 1}
38 num_material_classes = len(set(material_mapping.values()))
39
40 # DATASET
41 class SPTtoUTTDataset(torch.utils.data.Dataset):
42     def __init__(self, db_path, split='train', test_split=0.2, val_split=0.1, seed=42)
43     :
44         self.conn = sqlite3.connect(db_path)
45         df = pd.read_sql_query("SELECT * FROM paired_data", self.conn)
46         df['SPT_Curve'] = df['SPT_Curve'].apply(json.loads)
47         df['UTT_Curve'] = df['UTT_Curve'].apply(json.loads)
48         random.seed(seed)
49         self.train_set, self.val_set, self.test_set = self._split_data(df, test_split,
50         val_split)
51         self.data = {'train': self.train_set, 'val': self.val_set, 'test': self.
52         test_set}[split]
53
54     def _hash_curve(self, curve):
55         return hashlib.md5(np.array(curve).astype(np.float32).tobytes()).hexdigest()
56
57     def _split_data(self, df, test_split, val_split):
58         train_rows, val_rows, test_rows = [], [], []
59         grouped = df.groupby("Material")
60         for material, group in grouped:
61             group = group.copy()
62             group['SPT_ID'] = group['SPT_Curve'].apply(self._hash_curve)
63             group['UTT_ID'] = group['UTT_Curve'].apply(self._hash_curve)
64             spt_ids = np.array(group['SPT_ID'].unique())
65             utt_ids = np.array(group['UTT_ID'].unique())
66             random.shuffle(spt_ids)
67             random.shuffle(utt_ids)
68             spt_train, spt_val, spt_test = np.split(spt_ids, [int(len(spt_ids) * (1 -
69             test_split - val_split)), int(len(spt_ids) * (1 - test_split))])
70             utt_train, utt_val, utt_test = np.split(utt_ids, [int(len(utt_ids) * (1 -
71             test_split - val_split)), int(len(utt_ids) * (1 - test_split))])
72             for _, row in group.iterrows():
73                 sid, uid = row['SPT_ID'], row['UTT_ID']
74                 if sid in spt_train and uid in utt_train:
75                     train_rows.append(row)
76                 elif sid in spt_val and uid in utt_val:

```

```

72         val_rows.append(row)
73         elif sid in spt_test and uid in utt_test:
74             test_rows.append(row)
75         return pd.DataFrame(train_rows), pd.DataFrame(val_rows), pd.DataFrame(
76             test_rows)
77
78     def __len__(self):
79         return len(self.data)
80
81     def __getitem__(self, idx):
82         row = self.data.iloc[idx]
83         spt = torch.tensor(np.array(row['SPT_Curve']).T, dtype=torch.float32)
84         utt = torch.tensor(np.array(row['UTT_Curve']).T, dtype=torch.float32)
85         vec = torch.zeros(num_material_classes)
86         vec[material_mapping[row['Material']]] = 1.0
87         return spt, vec, utt, row['Material']
88
89 # MODEL
90 class CurveToCurveCNN(nn.Module):
91     def __init__(self, hidden_size, dropout_rate, activation):
92         super().__init__()
93         self.activation = activation
94         self.dropout = nn.Dropout(dropout_rate)
95
96         self.conv1 = nn.Conv1d(5, hidden_size, kernel_size=7, padding=3)
97         self.conv2 = nn.Conv1d(hidden_size, hidden_size, kernel_size=7, padding=3)
98         self.conv3 = nn.Conv1d(hidden_size, hidden_size * 2, kernel_size=7, padding=3)
99         self.conv4 = nn.Conv1d(hidden_size * 2, hidden_size * 2, kernel_size=7,
100             padding=3)
101         self.conv5 = nn.Conv1d(hidden_size * 2, hidden_size, kernel_size=7, padding=3)
102         self.conv6 = nn.Conv1d(hidden_size, hidden_size, kernel_size=7, padding=3)
103
104         self.head_10H2M = nn.Conv1d(hidden_size, 2, kernel_size=1)
105         self.head_13HMF = nn.Conv1d(hidden_size, 2, kernel_size=1)
106         self.head_15HM = nn.Conv1d(hidden_size, 2, kernel_size=1)
107
108     def forward(self, x, mat_onehot):
109         B, C, L = x.shape
110         mat = mat_onehot.unsqueeze(2).expand(B, -1, L)
111         x = torch.cat([x, mat], dim=1)
112
113         x = self._act(self.conv1(x))
114         x = self._act(self.conv2(x))
115         x = self._act(self.conv3(x))
116         x = self.dropout(self._act(self.conv4(x)))
117         x = self._act(self.conv5(x))
118         x = self._act(self.conv6(x))
119
120         mat_indices = torch.argmax(mat_onehot, dim=1)

```

```

119         out = torch.zeros(B, 2, L, device=x.device)
120
121         for i in range(B):
122             if mat_indices[i] == 0:
123                 out[i] = self.head_10H2M(x[i].unsqueeze(0)).squeeze(0)
124             elif mat_indices[i] == 1:
125                 out[i] = self.head_15HM(x[i].unsqueeze(0)).squeeze(0)
126             elif mat_indices[i] == 2:
127                 out[i] = self.head_13HMF(x[i].unsqueeze(0)).squeeze(0)
128         return out
129
130     def _act(self, layer):
131         return F.leaky_relu(layer) if self.activation == 'leaky_relu' else F.relu(
132             layer)
133
134 # TRAINING SETUP
135 train_loader = DataLoader(SPTtoUTTDataset(DB_PATH, 'train'), batch_size=params['
136     batch_size'], shuffle=True)
137 val_loader = DataLoader(SPTtoUTTDataset(DB_PATH, 'val'), batch_size=params['batch_size
138     '], shuffle=False)
139 test_loader = DataLoader(SPTtoUTTDataset(DB_PATH, 'test'), batch_size=1, shuffle=False
140     )
141
142 model = CurveToCurveCNN(params['hidden_size'], params['dropout'], params['activation'
143     ]).to(device)
144
145 optimizer = torch.optim.AdamW(model.parameters(), lr=0.0003, weight_decay=0.0005)
146 criterion = nn.MSELoss()
147 scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(optimizer, mode='min', patience
148     =5, factor=0.5)
149
150 best_val, wait, best_state = float('inf'), 0, None
151 train_curve, val_curve, history = [], [], []
152 total_time = 0
153
154 if os.path.exists(BEST_MODEL_PATH):
155     model.load_state_dict(torch.load(BEST_MODEL_PATH))
156
157 for epoch in range(1, 201):
158     start = time()
159     model.train()
160     running_loss = 0.0
161     for spt, mat, utt, _ in tqdm(train_loader, desc=f"Epoch {epoch:03d} [Train]",
162         leave=False):
163         spt, mat, utt = spt.to(device), mat.to(device), utt.to(device)
164         optimizer.zero_grad()
165         pred = model(spt, mat)
166         loss = criterion(pred, utt)
167         loss.backward()

```



```

161         optimizer.step()
162         running_loss += loss.item()
163
164     model.eval()
165     val_loss = 0.0
166     with torch.no_grad():
167         for spt, mat, utt, _ in tqdm(val_loader, desc=f"Epoch {epoch:03d} [Val]",
168             leave=False):
169             spt, mat, utt = spt.to(device), mat.to(device), utt.to(device)
170             pred = model(spt, mat)
171             val_loss += criterion(pred, utt).item()
172
173     val_loss /= len(val_loader)
174     current_lr = optimizer.param_groups[0]['lr']
175     elapsed = time() - start
176     total_time += elapsed
177     eta = (200 - epoch) * (total_time / epoch)
178
179     train_loss = running_loss / len(train_loader)
180     train_curve.append(train_loss)
181     val_curve.append(val_loss)
182     scheduler.step(val_loss)
183
184     history.append({"epoch": epoch, "train_loss": train_loss, "val_loss": val_loss, "
185         lr": current_lr})
186     pd.DataFrame(history).to_csv(f"{OUTPUT_DIR}/training_log_trial_{TRIAL_ID}.csv",
187         index=False)
188
189     if val_loss < best_val - 1e-5:
190         best_val = val_loss
191         best_state = model.state_dict()
192         torch.save(best_state, SAVE_MODEL_PATH)
193         with open(f"{OUTPUT_DIR}/best_model_meta.json", "w") as f:
194             json.dump({"epoch": epoch, "val_loss": best_val}, f, indent=2)
195     else:
196         wait += 1
197         if wait >= 10:
198             break
199
200 torch.save(best_state, SAVE_MODEL_PATH)
201
202 # EVALUATION
203 model.load_state_dict(torch.load(SAVE_MODEL_PATH))
204 model.eval()
205 results, per_material = [], {}
206
207 with torch.no_grad():
208     for i, (spt, mat, utt, name) in enumerate(test_loader):
209         spt, mat, utt = spt.to(device), mat.to(device), utt.to(device)

```

```

207     pred = model(spt, mat)
208     y_true = utt.squeeze().cpu().numpy().T
209     y_pred = pred.squeeze().cpu().numpy().T
210     results.append({"SampleID": i, "Material": name, "MAE": mean_absolute_error(
y_true, y_pred),
211                    "MSE": mean_squared_error(y_true, y_pred), "R2": r2_score(
y_true, y_pred)})
212     name = name if isinstance(name, str) else name[0]
213     per_material.setdefault(name, {"true": [], "pred": []})
214     per_material[name]["true"].extend(y_true[:, 1])
215     per_material[name]["pred"].extend(y_pred[:, 1])
216
217 pd.DataFrame(results).to_csv(f"{OUTPUT_DIR}/test_results_trial_{TRIAL_ID}.csv", index=
False)
218 pd.DataFrame([{"Material": k, "MAE": mean_absolute_error(v['true'], v['pred']),
219               "MSE": mean_squared_error(v['true'], v['pred']), "R2": r2_score(v['true
'], v['pred'])}]
220               for k, v in per_material.items())).to_csv(f"{OUTPUT_DIR}/
test_summary_per_material_trial_{TRIAL_ID}.csv", index=False)
221
222 plt.figure()
223 plt.plot(train_curve, label="Train Loss")
224 plt.plot(val_curve, label="Val Loss")
225 plt.title("Learning Curve")
226 plt.xlabel("Epoch")
227 plt.ylabel("Loss")
228 plt.legend()
229 plt.tight_layout()
230 plt.savefig(f"{OUTPUT_DIR}/learning_curve_trial_{TRIAL_ID}.png")
231 plt.close()
232
233 for mat, data in per_material.items():
234     plt.figure()
235     plt.scatter(data['true'], data['pred'], alpha=0.5)
236     plt.plot([min(data['true']), max(data['true'])], [min(data['true']), max(data['
true'])], 'r--')
237     plt.title(f"{mat} - True vs Predicted")
238     plt.xlabel("True Stress")
239     plt.ylabel("Predicted Stress")
240     plt.tight_layout()
241     plt.savefig(f"{OUTPUT_DIR}/scatter_{mat}_trial_{TRIAL_ID}.png")
242     plt.close()

```

Listing A.12: Program do precyzyjnego nauczania wybranego modelu sieci neuronowej CNN

```

1
2 import os
3 import pandas as pd
4 import torch

```

```

5 import torch.nn as nn
6 import torch.nn.functional as F
7 import matplotlib.pyplot as plt
8 import numpy as np
9 from sklearn.linear_model import LinearRegression
10 from scipy.signal import savgol_filter
11 from generate_re_rm_summary_v3 import analyze_and_plot_all
12
13 MATERIAL_CONSTANTS = {
14     '10H2M': {'S0': 19.46, 'L0': 19.93},
15     '13HMF': {'S0': 19.315, 'L0': 19.975},
16     '15HM': {'S0': 19.61, 'L0': 19.95},
17     'Other': {'S0': 19.462, 'L0': 19.952}
18 }
19
20 material_mapping = {'10H2M': 0, '13HMF': 2, '15HM': 1}
21 num_material_types = len(material_mapping)
22
23
24 class CurveToCurveCNN(nn.Module):
25     def __init__(self, hidden_size=512, dropout_rate=0.4, activation='relu'):
26         super().__init__()
27         self.activation = activation
28         self.dropout = nn.Dropout(dropout_rate)
29
30         self.conv1 = nn.Conv1d(5, hidden_size, kernel_size=7, padding=3)
31         self.conv2 = nn.Conv1d(hidden_size, hidden_size, kernel_size=7, padding=3)
32         self.conv3 = nn.Conv1d(hidden_size, hidden_size * 2, kernel_size=7, padding=3)
33         self.conv4 = nn.Conv1d(hidden_size * 2, hidden_size * 2, kernel_size=7,
padding=3)
34         self.conv5 = nn.Conv1d(hidden_size * 2, hidden_size, kernel_size=7, padding=3)
35         self.conv6 = nn.Conv1d(hidden_size, hidden_size, kernel_size=7, padding=3)
36
37         self.head_10H2M = nn.Conv1d(hidden_size, 2, kernel_size=1)
38         self.head_13HMF = nn.Conv1d(hidden_size, 2, kernel_size=1)
39         self.head_15HM = nn.Conv1d(hidden_size, 2, kernel_size=1)
40
41     def forward(self, x, mat_onehot):
42         B, C, L = x.shape
43         mat = mat_onehot.unsqueeze(2).expand(B, -1, L)
44         x = torch.cat([x, mat], dim=1)
45
46         x = self._act(self.conv1(x))
47         x = self._act(self.conv2(x))
48         x = self._act(self.conv3(x))
49         x = self.dropout(self._act(self.conv4(x)))
50         x = self._act(self.conv5(x))
51         x = self._act(self.conv6(x))
52

```

```

53     mat_indices = torch.argmax(mat_onehot, dim=1)
54     out = torch.zeros(B, 2, L, device=x.device)
55
56     for i in range(B):
57         if mat_indices[i] == 0:
58             out[i] = self.head_10H2M(x[i].unsqueeze(0)).squeeze(0)
59         elif mat_indices[i] == 1:
60             out[i] = self.head_15HM(x[i].unsqueeze(0)).squeeze(0)
61         elif mat_indices[i] == 2:
62             out[i] = self.head_13HMF(x[i].unsqueeze(0)).squeeze(0)
63         else:
64             raise ValueError("Invalid material index")
65
66     return out
67
68     def _act(self, layer):
69         return F.leaky_relu(layer) if self.activation == 'leaky_relu' else F.relu(
70             layer)
71
72 def get_user_input():
73     print("Please select the material type:")
74     for i, material in enumerate(material_mapping.keys()):
75         print(f"{i + 1}. {material}")
76     print(f"{len(material_mapping) + 1}. Other")
77     choice = input("Enter the number: ")
78
79     try:
80         choice = int(choice)
81         if 1 <= choice <= len(material_mapping):
82             selected = list(material_mapping.keys())[choice - 1]
83             return selected, material_mapping[selected]
84         else:
85             return "Other", -1
86     except ValueError:
87         return "Other", -1
88
89
90 def get_material_constants(material_label):
91     return MATERIAL_CONSTANTS.get(material_label, MATERIAL_CONSTANTS['Other'])
92
93
94 def generate_stress_strain_curves(utt_folder, output_folder, constants):
95     S0 = constants['S0']
96     L0 = constants['L0']
97     os.makedirs(output_folder, exist_ok=True)
98
99     for file_name in os.listdir(utt_folder):
100         if not file_name.endswith('.csv'):

```

```

101         continue
102
103         file_path = os.path.join(utt_folder, file_name)
104         df = pd.read_csv(file_path)
105
106         try:
107             h0_measured = int(file_name.split('_')[0])
108             correction = 500 / h0_measured
109         except Exception:
110             correction = 1.0
111
112         if 'UTT_Force' not in df.columns or 'UTT_Distance' not in df.columns:
113             continue
114
115         df['Stress'] = (df['UTT_Force'] * correction) / S0
116         df['Strain'] = df['UTT_Distance'] / L0
117
118         out_df = df[['Strain', 'Stress']]
119         output_path = os.path.join(output_folder, file_name)
120         out_df.to_csv(output_path, index=False)
121
122
123 def process_all_steps(root_directory, model_path, selected_material, material_index):
124     model = CurveToCurveCNN(hidden_size=512, dropout_rate=0.4, activation='relu')
125     model.load_state_dict(torch.load(model_path))
126     model.eval()
127
128     constants = get_material_constants(selected_material)
129
130     normalized_root = os.path.join(root_directory, 'Normalized')
131     process_all_files(root_directory, normalized_root, normalize_data)
132
133     utt_root = os.path.join(root_directory, 'UTT')
134     process_all_files(normalized_root, utt_root, generate_utt_data, model,
135                       material_index)
136
137     curves_root = os.path.join(root_directory, 'Curves')
138     generate_stress_strain_curves(utt_root, curves_root, constants)
139
140 def normalize_data(file_path, output_folder):
141     df = pd.read_csv(file_path, sep='\t', decimal=',', usecols=['Force', 'Distance'])
142     df['Force'] = df['Force'].clip(lower=0)
143     df['Distance'] = df['Distance'].clip(lower=0)
144     peak_idx = df['Force'].idxmax()
145     df = df.iloc[:peak_idx + 1].copy()
146     df['Normalized Force'] = df['Force'] / 1800.0
147     df['Normalized Distance'] = df['Distance'] / 1.8
148     out_df = df[['Normalized Force', 'Normalized Distance']]

```

```

149     os.makedirs(output_folder, exist_ok=True)
150     out_df.to_csv(
151         os.path.join(output_folder, 'normalized_' + os.path.basename(file_path)),
152         index=False, sep='\t', decimal=',',
153     )
154
155
156 def adaptive_sample_curve(curve, num_points=300):
157     curve = np.array(curve)
158     x, y = curve[:, 0], curve[:, 1]
159
160     if len(np.unique(x)) < 5 or np.allclose(x, x[0]):
161         idxs = np.linspace(0, len(curve) - 1, num_points).astype(int)
162         return curve[idxs].tolist()
163
164     try:
165         with np.errstate(divide='raise', invalid='raise'):
166             dy = np.gradient(y, x)
167             ddy = np.gradient(dy, x)
168
169             if (np.any(np.isnan(dy)) or np.any(np.isnan(ddy)) or
170                 np.any(np.isinf(dy)) or np.any(np.isinf(ddy))):
171                 raise FloatingPointError
172
173             weights = np.abs(dy) + 0.5 * np.abs(ddy)
174             weights = weights / np.sum(weights)
175             cdf = np.cumsum(weights)
176             cdf = cdf / cdf[-1]
177             uniform_samples = np.linspace(0, 1, num_points)
178             indices = np.searchsorted(cdf, uniform_samples)
179             indices = np.clip(indices, 0, len(curve) - 1)
180             return curve[indices].tolist()
181
182     except Exception:
183         idxs = np.linspace(0, len(curve) - 1, num_points).astype(int)
184         return curve[idxs].tolist()
185
186
187 def generate Utt_data(source_file, model, output_folder, material_index):
188     spt_df = pd.read_csv(source_file, sep='\t', decimal=',')
189     raw_curve = spt_df[['Normalized Distance', 'Normalized Force']].values
190     adapted = adaptive_sample_curve(raw_curve, num_points=300)
191     adapted = np.array(adapted)
192     features = adapted[:, [1, 0]].T
193     input_tensor = torch.tensor(features, dtype=torch.float32).unsqueeze(0)
194
195     if material_index != -1:
196         mat_tensor = F.one_hot(torch.tensor([material_index]), num_classes=
num_material_types).float()

```

```

197     else:
198         mat_tensor = torch.ones((1, num_material_types)) / num_material_types
199
200     with torch.no_grad():
201         utt_pred = model(input_tensor, mat_tensor)[0].T.numpy()
202
203     df = pd.DataFrame(utt_pred, columns=['UTT_Force', 'UTT_Distance'])
204     df['UTT_Force'] *= 13500.0
205     df['UTT_Distance'] *= 2.7
206     df['UTT_Distance'] = df['UTT_Distance'].clip(lower=0)
207     df.sort_values('UTT_Distance', inplace=True)
208     df.reset_index(drop=True, inplace=True)
209
210     os.makedirs(output_folder, exist_ok=True)
211     out_name = os.path.basename(source_file).replace('normalized_', '').replace('.txt',
212     , '_UTT.csv')
213     df.to_csv(os.path.join(output_folder, out_name), index=False)
214
215 def process_all_files(source_root, output_root, process_fn, model=None, material_index
    =None):
216     for dirpath, _, filenames in os.walk(source_root):
217         for filename in filenames:
218             if filename.endswith('.txt'):
219                 f_path = os.path.join(dirpath, filename)
220                 rel_path = os.path.relpath(dirpath, source_root)
221                 out_path = os.path.join(output_root, rel_path)
222                 if process_fn == generate_utt_data:
223                     process_fn(f_path, model, out_path, material_index)
224                 else:
225                     process_fn(f_path, out_path)
226                 print(f"Processed: {filename}")
227
228
229 if __name__ == "__main__":
230     root_directory = input("Enter the path to the root directory: ").strip().strip('"')
231     .strip('"')
232     model_path = "/Users/.../final_model_trial_1_retrained.pt"
233     selected_material, material_index = get_user_input()
234     process_all_steps(root_directory, model_path, selected_material, material_index)
235     curves_root = os.path.join(root_directory, "Curves")
236     analyze_and_plot_all(curves_root, selected_material)

```

Listing A.13: Główny program nowej sieci neuronowej CNN

```

1
2 import os
3 import pandas as pd
4 import matplotlib.pyplot as plt

```

```

5 import numpy as np
6 from sklearn.linear_model import LinearRegression
7 from sklearn.metrics import r2_score
8 from scipy.interpolate import interp1d
9 from scipy.optimize import brentq
10 from datetime import datetime
11
12 MATERIAL_CONSTANTS = {
13     '10H2M': {'S0': 19.46, 'L0': 19.93},
14     '13HMF': {'S0': 19.315, 'L0': 19.975},
15     '15HM': {'S0': 19.61, 'L0': 19.95},
16     'Other': {'S0': 19.462, 'L0': 19.952}
17 }
18
19 def elastic_fit_by_two_points(strain, stress):
20     best_delta = -np.inf
21     best_pair = (0, 1)
22     for i in range(len(stress) - 1):
23         delta = stress[i + 1] - stress[i]
24         if delta > best_delta:
25             best_delta = delta
26             best_pair = (i, i + 1)
27
28     i, j = best_pair
29     x1, x2 = strain[i], strain[j]
30     y1, y2 = stress[i], stress[j]
31     slope = (y2 - y1) / (x2 - x1)
32     intercept = y1 - slope * x1
33     return slope, intercept, 1.0, x1, x2
34
35 def calculate_Re_offset(strain_scaled, stress_scaled, slope_E, intercept_E, offset
=0.002, strain_raw=None, stress_raw=None):
36     def find_crossings(strain, stress):
37         offset_line = slope_E * (strain - offset) + intercept_E
38         diff = stress - offset_line
39         sign_change = np.where(np.diff(np.sign(diff)))[0]
40         intersections = []
41         for j in sign_change:
42             try:
43                 root = brentq(lambda eps: (
44                     interp1d(strain, stress)(eps) - (slope_E * (eps - offset) +
45                     intercept_E)
46                     ), strain[j], strain[j + 1])
47                 stress_val = interp1d(strain, stress)(root)
48                 intersections.append((root, stress_val))
49             except Exception:
50                 continue
51     return intersections

```



```

52 intersections_scaled = find_crossings(strain_scaled, stress_scaled)
53 intersections_raw = find_crossings(strain_raw, stress_raw) if strain_raw is not
None else []
54 all_candidates = intersections_scaled + intersections_raw
55
56 if all_candidates:
57     return max(all_candidates, key=lambda x: x[1])
58 else:
59     diff = stress_scaled - (slope_E * (strain_scaled - offset) + intercept_E)
60     min_idx = np.argmin(np.abs(diff))
61     return strain_scaled[min_idx], stress_scaled[min_idx]
62
63 def analyze_and_plot_all(curve_folder, material):
64     constants = MATERIAL_CONSTANTS[material]
65     S0, L0 = constants['S0'], constants['L0']
66     graphs_dir = os.path.join(curve_folder, 'Graphs')
67     os.makedirs(graphs_dir, exist_ok=True)
68
69     summary = []
70
71     for file in os.listdir(curve_folder):
72         if not file.endswith('.csv') or file == 'summary.csv':
73             continue
74
75         path = os.path.join(curve_folder, file)
76         df = pd.read_csv(path)
77         strain, stress = df['Strain'].values, df['Stress'].values
78         valid = ~np.isnan(stress)
79         strain_valid, stress_valid = strain[valid], stress[valid]
80
81         if len(strain_valid) > 10:
82             strain_valid = strain_valid[10:]
83             stress_valid = stress_valid[10:]
84         else:
85             continue
86
87         min_strain = np.min(strain_valid)
88         strain_valid = strain_valid - min_strain
89
90         strain_interp = np.linspace(strain_valid.min(), strain_valid.max(), 50)
91         stress_interp = interp1d(strain_valid, stress_valid, kind='linear')(
strain_interp)
92
93         slope_E, intercept_E, best_r2, fit_start, fit_end = elastic_fit_by_two_points(
strain_interp, stress_interp)
94         Re_strain, Re_stress = calculate_Re_offset(
95             strain_interp, stress_interp, slope_E, intercept_E,
96             strain_raw=strain_valid, stress_raw=stress_valid
97         )

```

```

98     Rm = np.max(stress_valid)
99
100    y_limit = Rm + 50
101    plt.figure(figsize=(10, 6))
102    plt.plot(strain_valid, stress_valid, label='Raw Curve', color='gray', alpha
=0.6)
103    plt.plot(strain_interp, stress_interp, label='Stress-Strain Curve', color='
blue', linewidth=2)
104    plt.plot(strain_interp, slope_E * strain_interp + intercept_E, '--', color='
orange', label='Elastic Fit')
105    plt.plot(strain_interp, slope_E * (strain_interp - 0.002) + intercept_E, ':',
color='red', label='0.2% Offset Line')
106    if Re_strain and Re_stress:
107        plt.scatter(Re_strain, Re_stress, color='black', label=f'Yield Strength {
Re_stress:.1f} MPa')
108        plt.scatter(strain_valid[np.argmax(stress_valid)], Rm, color='darkgreen',
label=f'Ultimate Strength {Rm:.1f} MPa')
109        plt.ylim(0, y_limit)
110        plt.xlabel('Strain')
111        plt.ylabel('Stress [MPa]')
112        plt.title(file)
113        plt.legend()
114        plt.grid(True)
115        plt.tight_layout()
116        graph_path = os.path.join(graphs_dir, file.replace('.csv', '.png'))
117        plt.savefig(graph_path)
118        plt.close()
119
120        summary.append({'Filename': file, 'Re': Re_stress, 'Rm': Rm})
121
122    summary_df = pd.DataFrame(summary)
123    avg_row = pd.DataFrame({'Filename': ['Average'], 'Re': [summary_df['Re'].mean()],
'Re': [summary_df['Rm'].mean()]})
124    summary_df = pd.concat([summary_df, avg_row], ignore_index=True)
125
126    folder_name = os.path.basename(os.path.dirname(os.path.normpath(curve_folder)))
127    timestamp = datetime.now().strftime("%Y-%m-%d_%H-%M")
128    summary_filename = f"summary_{folder_name}_{timestamp}.csv"
129    summary_path = os.path.join(curve_folder, summary_filename)
130    summary_df.to_csv(summary_path, index=False)
131    print(f"Saved summary to {summary_path}")

```

Listing A.14: Program do generowania wykresów UTT na podstawie danych z NN oraz wyznaczania własności materiału.

```

1
2 import sqlite3
3 import json
4 import torch

```

```

5 import torch.nn as nn
6 import torch.nn.functional as F
7 import numpy as np
8 import os
9 from sklearn.model_selection import train_test_split
10 from sklearn.metrics import accuracy_score, classification_report
11 from torch.utils.data import Dataset, DataLoader
12 from tqdm import tqdm
13
14 OUTPUT_DIR = '/Users/...'
15 os.makedirs(OUTPUT_DIR, exist_ok=True)
16
17
18 class SPTDatasetFromDB(Dataset):
19     def __init__(self, db_path, allowed_states, split='train', seed=42):
20         self.conn = sqlite3.connect(db_path)
21         self.samples = []
22
23         all_entries = []
24         cursor = self.conn.execute("SELECT SPT_Curve, Material, SPT_File FROM
paired_data")
25         for curve_str, material, spt_file in cursor:
26             if material not in allowed_states:
27                 continue
28             try:
29                 curve = np.array(json.loads(curve_str), dtype=np.float32).T
30                 if curve.shape[0] != 2:
31                     continue
32             except Exception:
33                 continue
34
35             label = 0 if material == allowed_states[0] else 1
36             all_entries.append((spt_file, curve, label))
37
38             unique_dict = {spt_file: (curve, label) for spt_file, curve, label in
all_entries if spt_file not in all_entries}
39             spt_files = list(unique_dict.keys())
40             train_val_files, test_files = train_test_split(spt_files, test_size=0.15,
random_state=seed)
41             train_files, val_files = train_test_split(train_val_files, test_size=0.15,
random_state=seed)
42
43             selected_files = {
44                 'train': set(train_files),
45                 'val': set(val_files),
46                 'test': set(test_files)
47             }.get(split, set())
48
49             for spt_file in selected_files:

```

```

50         curve, label = unique_dict[spt_file]
51         self.samples.append((curve, label, spt_file))
52
53     def __len__(self):
54         return len(self.samples)
55
56     def __getitem__(self, idx):
57         x, y, fname = self.samples[idx]
58         return torch.tensor(x), torch.tensor(y, dtype=torch.float32), fname
59
60
61 class SPTBinaryClassifier(nn.Module):
62     def __init__(self, input_channels=2, hidden_size=128, dropout_rate=0.2):
63         super().__init__()
64         self.conv1 = nn.Conv1d(input_channels, hidden_size, kernel_size=7, padding=3)
65         self.conv2 = nn.Conv1d(hidden_size, hidden_size, kernel_size=7, padding=3)
66         self.conv3 = nn.Conv1d(hidden_size, hidden_size, kernel_size=7, padding=3)
67         self.pool = nn.AdaptiveAvgPool1d(1)
68         self.dropout = nn.Dropout(dropout_rate)
69         self.fc = nn.Linear(hidden_size, 1)
70
71     def forward(self, x):
72         x = F.relu(self.conv1(x))
73         x = F.relu(self.conv2(x))
74         x = F.relu(self.conv3(x))
75         x = self.pool(x).squeeze(-1)
76         x = self.dropout(x)
77         return torch.sigmoid(self.fc(x))
78
79
80 def train_classifier_until_threshold(db_path, material, states, accuracy_target=0.95,
81                                     max_epochs=500):
82     train_set = SPTDatasetFromDB(db_path, states, split='train')
83     val_set = SPTDatasetFromDB(db_path, states, split='val')
84     test_set = SPTDatasetFromDB(db_path, states, split='test')
85
86     train_loader = DataLoader(train_set, batch_size=16, shuffle=True)
87     val_loader = DataLoader(val_set, batch_size=16)
88     test_loader = DataLoader(test_set, batch_size=1)
89
90     model = SPTBinaryClassifier()
91     optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
92     criterion = nn.BCELoss()
93
94     for epoch in range(max_epochs):
95         model.train()
96         running_loss = 0.0
97         for x, y, _ in train_loader:
98             y_pred = model(x).squeeze()

```

```

98         loss = criterion(y_pred, y)
99         optimizer.zero_grad()
100        loss.backward()
101        optimizer.step()
102        running_loss += loss.item()
103
104    model.eval()
105    y_true, y_pred = [], []
106    with torch.no_grad():
107        for x, y, _ in val_loader:
108            output = model(x).squeeze()
109            y_true.extend(y.numpy())
110            y_pred.extend(output.numpy())
111
112    acc = accuracy_score(y_true, np.round(y_pred))
113    avg_loss = running_loss / len(train_loader)
114    print(f"[{material}] Epoch {epoch + 1} - AVG Loss: {avg_loss:.4f} - Val
Accuracy: {acc:.4f}")
115
116    if acc >= accuracy_target and avg_loss <= 0.05:
117        print(f"Stable target reached at epoch {epoch + 1}: Acc {acc:.3f}, Loss {
avg_loss:.3f}")
118        break
119
120    print(f"\n[{material}] Test Set Evaluation:")
121    model.eval()
122    test_y_true, test_y_pred, mismatches = [], [], []
123    with torch.no_grad():
124        for x, y, fname in test_loader:
125            output = model(x).item()
126            pred = round(output)
127            test_y_true.append(y.item())
128            test_y_pred.append(output)
129            if pred != int(y.item()):
130                mismatches.append((fname, y.item(), output))
131
132    acc = accuracy_score(test_y_true, np.round(test_y_pred))
133    report = classification_report(test_y_true, np.round(test_y_pred), output_dict=
True)
134    return model, acc, mismatches, report
135
136
137 if __name__ == "__main__":
138     db_path = '/Users/.../re_rm_database_v2.3_small.db'
139     classifiers = {
140         "10H2M": ["J1", "D1"],
141         "15HM": ["T1", "H1"],
142         "13HMF": ["R1", "M3"],
143     }

```

```
144
145     for material, states in classifiers.items():
146         print(f"\nTraining classifier for: {material}")
147         model, acc, mismatches, report = train_classifier_until_threshold(
148             db_path=db_path,
149             material=material,
150             states=states,
151             accuracy_target=0.95
152         )
153
154         torch.save(model.state_dict(), os.path.join(OUTPUT_DIR, f"classifier_{material}
155             }2.pt"))
156
157         with open(os.path.join(OUTPUT_DIR, f"mismatches_{material}.txt"), "w") as f:
158             for fname, true_label, prob in mismatches:
159                 f.write(f"{fname} | True: {true_label} | Predicted prob: {prob:.3f}\n")
160
161         print(f"Final test accuracy for {material}: {acc:.2%}")
```

Listing A.15: Program do oceny stanu materiału z próbek SPT

Bibliografia

- [1] M. Abendroth and B. Kiefer. Characterisation of Shape Memory Alloys Using the Small Punch Test. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [2] Martin Abendroth and Meinhard Kuna. Determination of deformation and failure properties of ductile materials by means of the small punch test and neural networks. *Computational Materials Science*, 28(3-4 SPEC. ISS.):633–644, 2003.
- [3] Martin Abendroth and Stefan Soltysiak. Assessment of Material Properties by Means of the Small Punch Test. *Recent Trends in Fracture and Damage Mechanics*, pages 127–157, 2016.
- [4] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019.
- [5] J M Alegre, R Lacalle, I I Cuesta, and J A Álvarez. Different methodologies to obtain the fracture properties of metallic materials using pre-notched small punch test specimens. *Theoretical and Applied Fracture Mechanics*, 86:11–18, 2016.
- [6] L.F.M. Almeida, J.G.F. Peres, G.A. Martiniano, D.C.F. Ferreira, W.W. Bose Filho, and S.D. Franco. J-R curve evaluation using Small Punch Test and multiple specimen technique. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*, Santander, Spain, 9 2023. Universidad de Cantabria.
- [7] E. Altstadt, F. Bergner, and M. Houska. Use of the small punch test for the estimation of ductile-to-brittle transition temperature shift of irradiated steels. *Nuclear Materials and Energy*, 26(December 2020):100918, 2021.
- [8] E. Altstadt, H. E. Ge, V. Kuksenkov, M. Serrano, M. Houska, M. Lasan, M. Bruchhausen, J. M. Lape-tite, and Y. Dai. Critical evaluation of the small punch test as a screening procedure for mechanical properties. *Journal of Nuclear Materials*, 472:186–195, 2016.

- [9] E. Altstadt, M. Houska, I. Simonovski, M. Bruchhausen, S. Holmström, and R. Lacalle. On the estimation of ultimate tensile stress from small punch testing. *International Journal of Mechanical Sciences*, 136(November 2017):85–93, 2018.
- [10] G. Alvarez, A. Diaz, E. Martinez Paneda, F.J. Belzunce, and C. Rodrigues. Enhancing understanding of hydrogen assisted fracture in small punch test through computational modelling. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [11] G. Alvarez, M.M. Llera, F.J. Belzunce, and C. Rodrigues. Evolution of the Small Punch Test procedure to analyze the hydrogen embrittlement phenomenon. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [12] American Society for Nondestructive Testing. *Ultrasonic Testing Handbook, Vol. 7*. ASNT, Columbus, OH, 2007.
- [13] B. Arroyo, L. Andrea, J.A. Alvarez, F. Gutierrez-Solana, and E. Guilbert. Optimal steel times for ASTM F1624 standard methodology application to Small Punch Tests in aggressive environments. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [14] S. Arunkumar. Overview of Small Punch Test. *Metals and Materials International*, 26(6):719–738, 2020.
- [15] S. Arunkumar. Small Punch Creep Test: An Overview. *Metals and Materials International*, 2020(April), 2020.
- [16] ASTM International. ASTM E243 Standard Practice for Electromagnetic (Eddy Current) Examination of Copper and Copper Alloy Tubes. ASTM Standard, 2022.
- [17] Michal Besterci, Ferdinand Dobeš, Beáta Ballóková, Katarína Sülleiová, and Tibor Kvačkaj. Observation of anisotropy of creep fracture using small punch test for Al-Al 4C 3 system produced by equal channel angular pressing. *High Temperature Materials and Processes*, 30(3):205–210, 2011.
- [18] British Standards Institution. Metallic materials - Small punch test method, 2021.
- [19] British Standards Institution. *Metallic Materials. Uniaxial Creep Testing in Tension. Method of Test*. British Standards Institution, 2023.
- [20] M. Bruchhausen, S. Holmström, I. Simonovski, T. Austin, J. M. Lapetite, S. Ripplinger, and F. de Haan. Recent developments in small punch testing: Tensile properties and DBTT. *Theoretical and Applied Fracture Mechanics*, 86:2–10, 2016.

- [21] Jose Calaf-Chica, Pedro Miguel Bravo Díez, Mónica Preciado Calzada, and Daniel Ballorca-Juez. A systematic FEM analysis of the influence of mechanical properties in the reliability of the correlation methods in the small punch test. *International Journal of Mechanical Sciences*, 153-154(January):299–309, 2019.
- [22] Jose Calaf-Chica, Mario Sánchez Palomar, Pedro Miguel Bravo Díez, and Mónica Preciado Calzada. Deviations in yield and ultimate tensile strength estimation with the Small Punch Test: Numerical analysis of pre-straining and Bauschinger effect influence. *Mechanics of Materials*, 153(September 2020), 2021.
- [23] E. N. Campitelli, P. Spätig, R. Bonadé, W. Hoffelner, and M. Victoria. Assessment of the constitutive properties from small ball punch test: Experiment and modeling. *Journal of Nuclear Materials*, 335(3):366–378, 2004.
- [24] Yan Cao, Abdeliazim Mustafa Mohamed, Samira Moshtaghi, Morteza Taheri, and Mohammad Javad Torkamany. Investigation of creep behavior of Ni3Al-base superalloy by small punch creep. *Vacuum*, 187(October 2020):110101, 2021.
- [25] H. Chen, R. Yang, H. K. Al-Abedy, H. Li, W. Sun, and I. A. Jones. Characterisation of deformation process and fracture mechanisms of P91 steel at 600 °C in small punch tensile testing. *Materials Characterization*, 168(July), 2020.
- [26] Jianjun Chen, Young Wha Ma, and Kee Bong Yoon. Finite element study for determination of material's creep parameters from small punch test. *Journal of Mechanical Science and Technology*, 24(6):1195–1201, 2010.
- [27] José Calaf Chica, Pedro Miguel Bravo Díez, and Mónica Preciado Calzada. Improved correlation for elastic modulus prediction of metallic materials in the Small Punch Test. *International Journal of Mechanical Sciences*, 134(September):112–122, 2017.
- [28] N. J. Dunne, D. Leonard, C. Daly, F. J. Buchanan, and J. F. Orr. Validation of the small-punch test as a technique for characterizing the mechanical properties of acrylic bone cement. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 220(1):11–21, 2006.
- [29] Petr Dymáček, Ferdinand Dobeš, and Luboš Kloc. Determination of mechanical properties of metallic materials from very small volumes by means of small punch test. *Key Engineering Materials*, 741 KEM:116–121, 2017.
- [30] Petr Dymáček and Karel Milička. Creep small-punch testing and its numerical simulations. *Materials Science and Engineering A*, 510-511(C):444–449, 2009.
- [31] B.P. Eftink, M.E. Quintana, T.J. Romero, T.E. Steckley, P.L. Caccamise, S.A. Maloy, and T.A. Saleh. In hot cell shear punch testing of neutron irradiated steels. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors,

- 6th International Small Sample test Techniques Conference SSTT 2023. Universidad de Cantabria, 9 2023.
- [32] European Committee for Standardization. *Metallic materials - Charpy pendulum impact test - Part 1: Test method (ISO 148-1:2016)*. European Committee for Standardization, 2016.
- [33] Brian Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. *Cluster Analysis*. Wiley, 5 edition, 2011.
- [34] M. Fernández, C. Rodríguez, F. J. Belzunce, and T. E. García. Use of small punch test to estimate the mechanical properties of powder metallurgy products employed in the automotive industry. *Powder Metallurgy*, 58(3):171–177, 2015.
- [35] T. E. García, C. Rodríguez, F. J. Belzunce, and C. Suárez. Estimation of the mechanical properties of metallic materials by means of the small punch test. *Journal of Alloys and Compounds*, 582:708–717, 2014.
- [36] E Gdoutos. *Fracture of Nano and Engineering Materials and Structures: Proceedings of the 16th European Conference of Fracture, Alexandroupolis, Greece, July 3–7, 2006*. Springer Dordrecht, 1 2006.
- [37] Shyam R. Ghodke, Bijan K. Dutta, and P. V. Durgaprasad. Analytical development and experimental verification of empirical correlations to determine mechanical properties of copper alloys using small punch test data. *Fusion Engineering and Design*, 159(May), 2020.
- [38] V. L. Giddings, S. M. Kurtz, C. W. Jewett, J. R. Foulds, and A. A. Edidin. A small punch test technique for characterizing the elastic modulus and fracture behavior of PMMA bone cement used in total joint replacement. *Biomaterials*, 22(13):1875–1881, 2001.
- [39] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 11 2016.
- [40] A. L. Gurson. Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I—Yield Criteria and Flow Rules for Porous Ductile Media. *Journal of Engineering Materials and Technology*, 99(1):2–15, 1 1977.
- [41] Jeong Soo Ha and Eric Ficury. Small Punch Tests on Steels for Steam Power Plant (I). *KSME International Journal*, I(I), 1998.
- [42] Peter Hähner, Celal Soyarslan, Betül Gülçimen Çakan, and Swantje Bargmann. Determining tensile yield stresses from Small Punch tests: A numerical-based scheme. *Materials and Design*, 182:107974, 2019.
- [43] S Holmström, Y Li, P Dymacek, E Vacchieri, S P Je, R J Lancaster, D Omacht, Z Kubon, E Anelli, J Rantala, A Tonti, S Komazaki, M Bruchhausen, R C Hurst, P Hähner, M Richardson, and D Andres. Creep strength and minimum strain rate estimation from Small Punch Creep tests. *Materials Science & Engineering*, 731(April):161–172, 2018.

- [44] S. Holmström, I. Simonovski, D. Baraldi, M. Bruchhausen, E. Altstadt, and R. Delville. Developments in the estimation of tensile strength by small punch testing. *Theoretical and Applied Fracture Mechanics*, 101(November 2018):25–34, 2019.
- [45] Peter J Huber. *Robust Statistics*. Wiley, New York, 1981.
- [46] J D Hunter and Contributors. *matplotlib: Visualization with Python*, 2024.
- [47] T. H. Hyde, M. Stoyanov, W. Sun, and C. J. Hyde. On the interpretation of results from small punch creep tests. *Journal of Strain Analysis for Engineering Design*, 45(3):141–164, 2010.
- [48] T. H. Hyde, W. Sun, and J. A. Williams. Requirements for and use of miniature test specimens to provide mechanical and creep properties of materials: A review. *International Materials Reviews*, 52(4):213–255, 2007.
- [49] Industrial Physics. Microhardness Testing Explained: Vickers and Knoop Testing. <https://industrialphysics.com/knowledgebase/articles/microhardness-testing-explained-vickers-knoop-testing/>, 2024.
- [50] International Organization for Standardization. *International Standard ISO 6892-1 Metallic materials — Tensile testing*, volume 2009. International Organization for Standardization, 2009.
- [51] International Organization for Standardization. ISO 14577-1:2015 Instrumented indentation test for hardness and materials parameters — Part 1: Test method. ISO Standard, 7 2014.
- [52] International Organization for Standardization. ISO 6507-1:2023 Metallic materials — Vickers hardness test — Part 1: Test method. ISO Standard, 10 2023.
- [53] G. A. Jackson, W. Sun, and D. G. McCartney. The application of the small punch tensile test to evaluate the ductile to brittle transition of a thermally sprayed conical coating. *Key Engineering Materials*, 734 KEM:144–155, 2017.
- [54] David J. Jaekel, Daniel W. MacDonald, and Steven M. Kurtz. Characterization of PEEK biomaterials using the small punch test. *Journal of the Mechanical Behavior of Biomedical Materials*, 4(7):1275–1282, 2011.
- [55] Adam Janča, Jan Siegl, and Petr Haušild. Small punch test evaluation methods for material characterisation. *Journal of Nuclear Materials*, 481:201–213, 2016.
- [56] Jun Young Jeon, Yun Jae Kim, Sa Yong Lee, and Jin Weon Kim. Extracting ductile fracture toughness from small punch test data using numerical modeling. *International Journal of Pressure Vessels and Piping*, 139-140:204–219, 2016.
- [57] T. Jiang, J. Zhong, and K. Guan. Correlation and extrapolation of small punch creep test data for Incoloy 800H and 347H heat-resistant alloys. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.

- [58] T. Jiang, J. Zhong, and K. Guan. Investigation of deformation mechanism and creep behavior of 347H stainless steel in small punch creep test. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [59] David Jiles. *Introduction to Magnetism and Magnetic Materials*. CRC Press, 9 2015.
- [60] Stephen C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.
- [61] Maciej Kaliciak. Dane replikacji dla: Automatyzacja oceny stanu materiału z wykorzystaniem metody SPT, 8 2025.
- [62] Maciej Kaliciak and Tadeusz Uhl. Small Punch Test (SPT) - Najnowsze trendy współczesnej diagnostyki instalacji przemysłowych. *Chemia przemysłowa*, pages 43–48, 2021.
- [63] J. Kameda. A kinetic model for ductile-brittle fracture mode transition behavior. *Acta Metallurgica*, 34(12):2391–2398, 1986.
- [64] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, New York, 2005.
- [65] Jeong Hwan Kim, Uiyeong Ro, Hoomin Lee, Seok Jun Kang, Byung Ho Lee, and Moon Ki Kim. A direct assessment of creep life based on small punch creep test. *Theoretical and Applied Fracture Mechanics*, 104(September):102346, 2019.
- [66] J. Kłaput. Application of small punch test method in studies of the 14MoV63 steel before and after revitalisation. *Archives of Metallurgy and Materials*, 60(1):47–50, 2015.
- [67] Yasutomo Koga, Yoshihiko Arao, and Masatoshi Kubouchi. Application of small punch test to lifetime prediction of plasticized polyvinyl chloride wire. *Polymer Degradation and Stability*, 171:109013, 2020.
- [68] R. Kopriva, M. Adamech, J. Petzova, L. Kander, and I. Schnablova. Proficiency testing of material parameters assessment for SPT methodology precision increase. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [69] Josef Krautkrämer and Herbert Krautkrämer. *Ultrasonic Testing of Materials*. Springer-Verlag, Berlin, Heidelberg, 4th edition, 1990.
- [70] Petr Kubík, František Šebek, Jindřich Petruška, Jiří Hůlka, Namsu Park, and Hoon Huh. Comparative investigation of ductile fracture with 316L austenitic stainless steel in small punch tests: Experiments and simulations. *Theoretical and Applied Fracture Mechanics*, 98(October):186–198, 2018.

- [71] Kundan Kumar, Arun Pooleery, K. Madhusoodanan, R. N. Singh, J. K. Chakravartty, R. S. Shrivastaw, B. K. Dutta, and R. K. Sinha. Evaluation of ultimate tensile strength using Miniature Disk Bend Test. *Journal of Nuclear Materials*, 461(December 2014):100–111, 2015.
- [72] R.J. Lancaster, N.C. Barnard, P.E. May, E.E. Sackett, and S.P. Jeffs. Assessing the Mechanical Behavior of an Additive Manufactured Nickel Based Superalloy Using Alternative Small Scale Test Techniques. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [73] Robert Lancaster, Gareth Davies, Henry Illsley, Spencer Jeffs, and Gavin Baxter. Structural integrity of an electron beam melted titanium alloy. *Materials*, 9(6), 2016.
- [74] Nicolas Leclerc, Ali Khosravani, Sepideh Hashemi, Daniel B. Miracle, and Surya R. Kalidindi. Correlation of Measured Load-Displacement Curves in Small Punch Tests with Tensile Stress-Strain Curves. *Acta Materialia*, 204:116501, 2021.
- [75] Jong Min Lee, Jin Ha Hwang, Yun Jae Kim, and Jin Weon Kim. Predicting ductile fracture of cracked pipes using small punch test data. *European Journal of Mechanics, A/Solids*, 87(January):104211, 2021.
- [76] Taeksang Lee, Hoomin Lee, Seok Jun Kang, Fahim Ahmed Ibupoto, Jeong Min Lee, Jong Hoon Lee, Bum Joon Kim, Jae Boong Choi, Siyeon Bae, and Moon Ki Kim. Small punch test and simulation of HR3C steel. *Journal of Mechanical Science and Technology*, 32(7):3115–3121, 2018.
- [77] Y. Li, P. Dymacek, K. Guan, F. Yu, M. Sun, and F. Zhang. Comments on standard EN-10371 and proposal for standard follow up. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [78] T. Linse, M. Kuna, J. Schuhknecht, and H. W. Viehrig. Usage of the small-punch-test for the characterisation of reactor vessel steels in the brittle-ductile transition region. *Engineering Fracture Mechanics*, 75(11):3520–3533, 2008.
- [79] M.M. Llera, G. Alvarez, S. Otero, A. Zafra, I.I. Cuesta, and C. Rodrigues. Effect of post-weld treatment on the hydrogen embrittlement of CrMoV steel welded joints using Small Punch Test. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [80] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [81] Amirreza Lotfolahpour, Nasser Soltani, Mehdi Ganjiani, and Davoud Baharlouei. Parameters identification and validation of plastic-damage model of 304 stainless steel by small punch test at ambient temperature. *Engineering Fracture Mechanics*, 200(February):64–74, 2018.

- [82] G. E. Lucas. Review of small specimen test techniques for irradiation testing. *Metallurgical Transactions A*, 21(5):1105–1119, 1990.
- [83] Tao Ma, Bin Zhang, Li Yi Wang, Zhu Man Song, Xue Mei Luo, Chang Sheng Liu, and Guang Ping Zhang. Anisotropy of small punch creep performance of selective laser melted GH4169 at 650 °C. *Materials Science and Engineering A*, 806, 3 2021.
- [84] Saverio Maietta, Teresa Russo, Roberto De Santis, Dante Ronca, Filomena Riccardi, Michelina Catauro, Massimo Martorelli, and Antonio Gloria. Further theoretical insight into the mechanical properties of polycaprolactone loaded with organic-inorganic hybrid fillers. *Materials*, 11(2):1–8, 2018.
- [85] Jonathan Mak, Richard Wuhner, Sam Humphries, Norman Booth, Greg Heness, Wing Yiu Yeung, Tao Wei, Jining Qin, Qiubao Ouyang, and Di Zhang. Small punch test of LC4/SiCp metal matrix composites. *Advanced Materials Research*, 123-125:439–442, 2010.
- [86] Xinyuan Mao and Hideaki Takahashi. Development of a further-miniaturized specimen of 3 mm diameter for tem disk ($\varnothing$ 3 mm) small punch tests. *Journal of Nuclear Materials*, 150(1):42–52, 1987.
- [87] E. Martínez-Pañeda, I. I. Cuesta, I. Peñuelas, A. Díaz, and J. M. Alegre. Damage modeling in Small Punch Test specimens. *Theoretical and Applied Fracture Mechanics*, 86:51–60, 2016.
- [88] J. McNaney, G. E. Lucas, and G. R. Odette. Application of ball punch tests to evaluating fracture mode transition in ferritic steels. *Journal of Nuclear Materials*, 179-181(PART 1):429–433, 1991.
- [89] W Q Meeker and L A Escobar. *Statistical Methods for Reliability Data*. Wiley Series in Probability and Statistics. Wiley, 2014.
- [90] Paul Meißner, Tom Hoppe, and Thomas Vietor. Comparative Study of Various Neural Network Types for Direct Inverse Material Parameter Identification in Numerical Simulations. *Applied Sciences*, 12(24):12793, 12 2022.
- [91] American Society for Metals and A S M International T A T T . ASM Handbook. Volume 12, Fractography, 1987.
- [92] Metinvest B.V. METINVEST Steel grades, 2025.
- [93] Minitab. Minitab 21 Support: Statistical Modeling, 2023.
- [94] T. Misawa, T. Adachi, M Saito, and Y. Hamaguchi. Small punch tests for evaluating ductile-brittle of irradiated ferritic steels. *Journal of Nuclear Materials*, 150:749–753, 1988.
- [95] Toshiya Nakata, Shin ichi Komazaki, Yutaka Kohno, and Hiroyasu Tanigawa. Development of a small punch testing method to evaluate the creep property of high Cr ferritic steel: Part I-Effect of atmosphere on creep deformation behavior. *Materials Science and Engineering A*, 666:54–60, 2016.

- [96] NDT Resource Center and Iowa State University. Eddy Current Testing Overview. <https://www.nde-ed.org/NDETechniques/EddyCurrent/eddycurrenttesting.xhtml>, 2024.
- [97] NDT Resource Center and Iowa State university. Ultrasonic Testing Overview. <https://www.nde-ed.org/NDETechniques/Ultrasonics/index.xhtml>, 2024.
- [98] NumPy Developers. NumPy v1.26 Manual, 2024.
- [99] D. Omacht, Y. Li Wang, B. Wang, M. Sun, F. Yu, F. Zhang, and W. Sun. Mechatronics design and architecture of evaluation software for a portable instrumented indentation system PIIS3000™. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 6 2023.
- [100] S. Oterro, G. Alvarez, M.M. Llera, F.J. Belzunce, and C. Rodriguez. Use of Small Punch Test (SPT) in the fatigue characterization of S355 structural steel. In Roberto Lacalle, J. Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*, Santander, Spain, 9 2023. Universidad de Cantabria.
- [101] Huayong Pan, Jian Peng, Xiangxuan Geng, Mingxuan Gao, and Xinting Miao. Prediction of mechanical properties for typical pressure vessel steels by small punch test combined with machine learning. *International Journal of Pressure Vessels and Piping*, 206, 12 2023.
- [102] pandas Development Team. pandas Documentation, 2024.
- [103] A.V. Pandey, V. Karthik, A. Kolhatkar, V. Shankar, and R. Divakar. Evaluation of material deformation in small punch fatigue with DIC technique. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [104] R.V. Prakash. Overview of the study on fatigue response of an Inconel alloy through Cyclic Ball Indentation and Cyclic Small Punch Testing. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [105] Rhys Pullin, Ifan Jenkins, Anghel Cernescu, and Allen Edwards. Equivalent biaxial strain evaluation in small punch testing using acoustic emission. *Journal of Strain Analysis for Engineering Design*, 56(3):173–180, 2020.
- [106] Python Software Foundation. The Python Standard Library: datetime — Basic date and time types, 2024.
- [107] Python Software Foundation. The Python Standard Library: hashlib — Secure hashes and message digests, 2024.

- [108] Python Software Foundation. The Python Standard Library: json — JSON encoder and decoder, 2024.
- [109] Python Software Foundation. The Python Standard Library: os — Miscellaneous operating system interfaces, 2024.
- [110] Python Software Foundation. The Python Standard Library: random — Generate pseudo-random numbers, 2024.
- [111] Python Software Foundation. The Python Standard Library: sqlite3 — DB-API 2.0 interface for SQLite databases, 2024.
- [112] Python Software Foundation. The Python Standard Library: time — Time access and conversions, 2024.
- [113] PyTorch Contributors. torch.optim.AdamW — PyTorch 2.1 documentation. <https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html>, 2023.
- [114] PyTorch Developers. PyTorch Documentation, 2024.
- [115] Ma Quanjin, M. R.M. Rejab, Qayyum Halim, M. N.M. Merzuki, and M. A.H. Darus. Experimental investigation of the tensile test using digital image correlation (DIC) method. In *Materials Today: Proceedings*, volume 27, pages 757–763. Elsevier Ltd, 1 2020.
- [116] W. D. Ray and J. D. Jobson. Applied Multivariate Data Analysis: Vol. 2, Categorical and Multivariate Methods. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 156(3):506, 1993.
- [117] C. Rodríguez, D. Arencón, J. Belzunce, and M. Ll MasPOCH. Small punch test on the analysis of fracture behaviour of PLA-nanocomposite films. *Polymer Testing*, 33:21–29, 2014.
- [118] C. Rodríguez, I. I. Cuesta, M. L.L. MasPOCH, and F. J. Belzunce. Application of the miniature small punch test for the mechanical characterization of polymer materials. *Theoretical and Applied Fracture Mechanics*, 86:78–83, 2016.
- [119] C. Rodríguez, J. García Cabezas, E. Cárdenas, F. J. Belzunce, and C. Betegón. Mechanical properties characterization of heat-affected zone using the small punch test. *Welding Journal (Miami, Fla)*, 88(9):188–192, 2009.
- [120] Sheldon M Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press, 5th edition, 2014.
- [121] Y. Ruan, P. Spätig, and M. Victoria. Assessment of mechanical properties of the martensitic steel EUROFER97 by means of punch tests. *Journal of Nuclear Materials*, 307-311(1 SUPPL.):236–239, 2002.
- [122] David Sánchez-Ávila, Alberto Orozco-Caballero, Elkin Martínez, Luis Portolés, Rafael Barea, and Fernando Carreño. High-accuracy compliance correction for nonlinear mechanical testing: Improving Small Punch Test characterization. *Nuclear Materials and Energy*, 26(May 2020), 2021.

- [123] M. L. Saucedo-Muñoz, V. Miranda-Lopez, S. Komazaki, and V. M. Lopez-Hirata. Relation of small punch creep test properties with microstructure changes for an ASTM A387 Cr–Mo steel. *Materials Science and Engineering A*, 761(April):138033, 2019.
- [124] Scikit-learn Developers. Scikit-learn User Guide, 2024.
- [125] SciPy Developers. SciPy v1.13.0 Reference Guide, 2024.
- [126] V. Shah, J. A.W. van Dommelen, E. Altstadt, A. Das, and M. G.D. Geers. Brittle-ductile transition temperature of recrystallized tungsten following exposure to fusion relevant cyclic high heat load. *Journal of Nuclear Materials*, 541, 2020.
- [127] A Shekhter, A B L Croker, A K Hellier, C J Moss, and S P Ringer. Towards the correlation of fracture toughness in an ex-service power generating rotor. *International Journal of Pressure Vessels and Piping*, 77:113–116, 2000.
- [128] Jan Siegl, Petr Haušild, Adam Janča, and Radim Kopřiva. Fractographic Aspects of Small Punch Test Results. *Procedia Materials Science*, 3:912–917, 2014.
- [129] Jan Siegl, Petr Haušild, Adam Janča, Radim Kopřiva, and Miloš Kytka. Characterisation of mechanical properties by small punch test. *Key Engineering Materials*, 606:15–18, 2014.
- [130] Jagjit Singh, N. K. Sharma, and Satbir S. Sehgal. SMALL PUNCH TESTING: AN ALTERNATIVE TESTING TECHNIQUE to EVALUATE TENSILE BEHAVIOR of CORTICAL BONE. *Journal of Mechanics in Medicine and Biology*, 17(6):1–14, 2017.
- [131] S. Singh, O. Santana-Pérez, C. Rodriguez, K. Oksman, and M. LI Maspoeh. Mechanical behaviour of poly(Lactic acid)/cellulose nanocrystal nanocomposites: A comparative study between conventional tensile test and small punch test. *Express Polymer Letters*, 14(12):1127–1136, 2020.
- [132] P H A Sneath, P H A Sneath, R R Sokal, and U R R Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. A Series of books in biology. W. H. Freeman, 1973.
- [133] Ming Song, Xuyang Li, Yuguang Cao, Ying Zhen, Jiru Zhong, and Kaishu Guan. Determination of elastoplastic properties of in-service pipeline steel based on backpropagation neural network and small punch test. *International Journal of Pressure Vessels and Piping*, 190(January), 2021.
- [134] J. Špička, L. Kander, and P. Čížek. Neural network utilization for evaluation of the steel material properties. *Ubiquity Proceedings*, 1(S1):45, 9 2018.
- [135] Stresstech. Barkhausen Noise Analysis. <https://www.stresstech.com/knowledge/non-destructive-testing-methods/barkhausen-noise-analysis/>, 2024.
- [136] H. Sun, Y. Ma, M. Yang, J. Zhong, and K. Huan. Evaluation of texture strengthening effect if pure zirconium sheet by small punch test. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.

- [137] tqdm Developers. tqdm: A Fast, Extensible Progress Bar for Python and CLI, 2024.
- [138] TS Industrial Corporation Limited. ASME SA516 Klasa 70, 4 2023.
- [139] V. Tvergaard and A. Needleman. Analysis of the cup-cone fracture in a round tensile bar. *Acta Metallurgica*, 32(1):157–169, 1 1984.
- [140] Urząd Dozoru Technicznego. Wytoczne UDT nr 1/2015, 2015.
- [141] U.S. Naval Academy. EN380 Naval Materials Science and Engineering Course Notes. Technical report, U.S. Naval Academy, 2022.
- [142] E. Vacchieri, L. Bruzzzone, E. Poggio, and A. Sanguineti. Application of small punch creep for creep residual life and life extension of gas turbine blades and vanes. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [143] H.B. van der Merwe, D.G. Hattingh, D. Bernard, and I. Wedderburn. Room and elevated temperature Small Punch Static Testing and analysis using empirical and numerical methods. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [144] VIRGAMET Sp. z o.o. Stal kotłowa do pracy w podwyższonych temperaturach, 2025.
- [145] Q. Wang, M. Wang, T. Xu, J. Zhong, and K. Guan. Determination of Fracture Toughness of Metallic Materials by Small Punch Test with a new type of slot structure. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [146] Wulin Wang, Jiru Zhong, Xiaocheng Zhang, Tao Jiang, and Kaishu Guan. Study of estimation of ductile-brittle transition temperature using U-notched small punch test specimens. *Theoretical and Applied Fracture Mechanics*, 108(December 2019), 2020.
- [147] T. Wei, D. G. Carr, E. Budzakoska, W. Payten, R. P. Harrison, and M. I. Ripley. Assessment of the fracture toughness of 6061 aluminium by the small punch test and finite element analysis. *Materials Forum*, 30:39–44, 2006.
- [148] Cheng Wen, Tong Xu, and Kaishu Guan. Correlation factor study of small punch creep test and its life prediction. *Materials*, 9(10), 2016.
- [149] Wu Wen, Xiaozhe Jin, Hao Liu, and Wei Sun. Determination of creep damage properties from small punch creep tests considering pre-straining effect using an inverse approach. *Mechanics of Materials*, 139(April), 2019.

- [150] H. Wilcox. Evaluation of Mechanical Yield Properties of 17-4PH using Small punch and sub-size Tensile Testing. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [151] Rand R Wilcox. *Introduction to Robust Estimation and Hypothesis Testing*. Academic Press, 3 edition, 2012.
- [152] B Wilshire and P J Scharning. Long-term creep life prediction for a high chromium steel. *Scripta Materialia*, 56:701–704, 2007.
- [153] Sisheng Yang, Lin Xue, and Xiang Ling. Characterisation of low-temperature creep deformation and primary creep stage of TA2 using small punch creep test. *Engineering Fracture Mechanics*, 205(October 2018):1–9, 2019.
- [154] C. F. Yao and Y. Dai. DBTT shift of Optifer-IX, Eurofer 97 and MA956 steels after irradiation evaluated with small punch tests. *Journal of Nuclear Materials*, 544:152725, 2021.
- [155] Bo Zhao, Jing Guo, Yuxin Yu, Xuetao Zhang, Tong Xu, and Binan Shou. A study on small punch test to characterize mechanical property of aged high-density polyethylene (HDPE). *Key Engineering Materials*, 734 KEM:104–115, 2017.
- [156] J. Zhong, Z. He, K. Guan, and J. Tao. Investigation on regression model for the force of small punch test using machine learning. In Roberto Lacalle, J Alberto Alvares, Borja Arroyo, Federico Gutierrez-Solana, Sergio Cicero, Marta Serrano, and David Andres, editors, *6th International Small Sample test Techniques Conference SSTT 2023*. Universidad de Cantabria, 9 2023.
- [157] Jiru Zhong, Tong Xu, Kaishu Guan, and Jerzy Szpunar. A procedure for predicting strength properties using small punch test and finite element simulation. *International Journal of Mechanical Sciences*, 152(November 2018):228–235, 2019.
- [158] Zjednoczenie Hutnictwa Żelaza i Stali. Blachy kotłowe ze stali 18 CuNMT BN-66/0642-12, 1966.