

# **Spis treści**

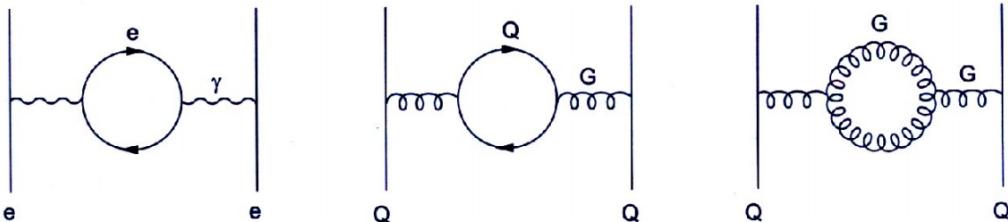
<b>1</b>	<b>Fizyka dżetów cząstek</b>	<b>2</b>
3	1.1 Chromodynamika kwantowa . . . . .	2
4	1.2 Plazma kwarkowo-gluonowa . . . . .	3
5	1.3 Dżety . . . . .	4
6	1.4 Dżety $b$ . . . . .	4
7	1.5 Eksperyment ALICE . . . . .	5
<b>2</b>	<b>Uczenie maszynowe</b>	<b>9</b>
9	2.1 Wzmacniane drzewa decyzyjne . . . . .	9
10	2.2 Sieci neuronowe . . . . .	10
11	2.3 Dyskusja użycia dwóch algorytmów . . . . .	15
<b>3</b>	<b>Dane</b>	<b>16</b>
<b>4</b>	<b>Analiza</b>	<b>19</b>
14	4.1 Dobór metryki . . . . .	19
15	4.2 Wyniki dla zmiennych SV . . . . .	22
16	4.3 Wyniki dla zmiennych constit . . . . .	22

# 17 1 Fizyka dżetów cząstek

## 18 1.1 Chromodynamika kwantowa

19 Chromodynamika kwantowa (ang. *Quantum Chromodynamics – QCD*) to kwantowa teoria  
20 pola opisująca oddziaływanie silne [1]. Wprowadza ona dla kwarków nową liczbę kwantową  
21 nazywaną kolorem lub ładunkiem kolorowym, który jest odpowiednikiem ładunku elektrycznego  
22 w elektrodynamice kwantowej (ang. *Quantum Electrodynamics – QED*), ale w przeciwnieństwie  
23 do niego może przyjmować 3 różne wartości (i trzy antywartości dla antykwarków). Elementarne  
24 oddziaływanie w obu teoriach przenoszone są przez bezmasowe bozony pośredniczące: w *QED*  
25 jest to elektrycznie obojętny foton a w *QCD* gluony, które występują w 8 odmianach i są  
26 kolorowo naładowane, przez co możliwe jest oddziaływanie zachodzące między dwoma gluonami.  
27 Kwarki i gluony zbiorczo nazywane są partonami.

28 Próżnia, w rozumieniu klasycznym będąca zupełnie pusta, w teoriach kwantowych wypeł-  
29 niona jest pojawiającymi i znikającymi wirtualnymi cząstkami. Cząstki te ekranują ładunek  
30 próbny umieszczony w kwantowej próżni, wywołując zjawisko polaryzacji próżni (analogiczne  
31 do polaryzacji dielektryków), które efektywnie zmniejsza pole wytwarzane przez ten ładunek.  
32 Siła tego efektu zależy od liczby ekranujących cząstek, czyli pośrednio od skali odległości. Skala  
33 ta wyznaczona jest przez długości fali próbującej cząstki, zatem także jej energię (im więk-  
34 sza energia, tym mniejsza długość fali i mniejsza ilość ekranujących cząstek obserwowanych  
35 w pobliżu rzeczywistego ładunku, zatem tym słabszy efekt ekranowania i większy efektywny  
36 ładunek). Prowadzi to do zależnej od energii stałej sprzężenia  $\alpha$ , którą nazywamy efektywną  
37 lub biegącą stałą sprzężenia (ang. *running coupling constant*).



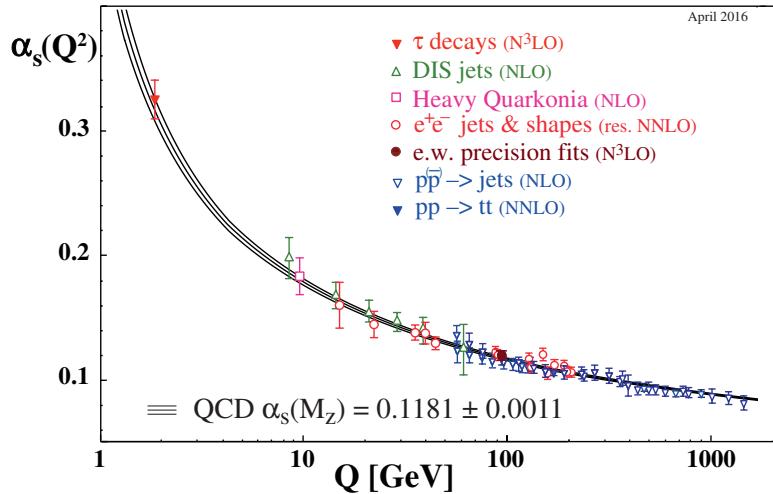
Rysunek 1: Diagramy Feynmana opisujące polaryzację próżni w *QED* (lewy) i *QCD* (środkowy i prawy). Rysunki lewy i środkowy są swoimi odpowiednikami w tych dwóch teoriach, natomiast prawy, w którym oddziałują jedynie bozony pośredniczące nie ma swojego odpowiednika w *QED*. Źródło: [1]

38 Zarówno pary elektron-pozyton jak i kwark-antykwark działają ekranującą kolejno ładu-  
39 nek elektryczny i kolorowy. Jednak jak zostało to już wspomniane, w przypadku *QCD* możliwe  
40 jest także samooddziałanie gluonów, przez co dopuszczalne są diagramy Feynmana jak ten  
41 przedstawiony na Rys. 1 po prawej. Pętle gluonowe działają anty-ekranującą – zwiększą efek-  
42 tywną wartość silnej stałej sprzężenia, ponadto jest to efekt dominujący nad przyczynkiem od  
43 par kwark-antykwark, co sprawia że zależność biegącej stałej sprzężenia w *QCD* jest odwrotna  
44 i dużo silniejsza niż w przypadku *QED*. Wartość  $\alpha_{em}$  maleje od wartości  $\frac{1}{128}$  przy energiach ok.  
45 90 GeV do  $\frac{1}{137}$  przy energii bliskiej zeru, co oznacza zmianę o kilka procent. Tymczasem  $\alpha_S$   
46 rośnie w miarę zbliżania się do niskich energii od wartości  $\alpha_S \lesssim 0.1$  dla  $E \gtrsim 100$  GeV do  $\alpha_S > 1$   
47 dla energii poniżej 200 MeV (por. Rys. 2). Prowadzi to do dwóch zjawisk charakterystycznych  
48 dla chromodynamiki kwantowej:

- 49 • asymptotyczna swoboda (ang. *asymptotic freedom*) [2], [3] – dla wysokich energii ( $\gtrsim 100$   
50 GeV) silna stała sprzężenia jest mała (w tym zakresie energii możliwe jest stosowanie

51 rachunku perturbacyjnego) i kwarki wewnątrz hadronów zachowują się jak cząstki quasi-  
52 swobodne.

- 53 • uwiecznienie koloru (ang. *colour confinement*) – przy zwiększaniu odległości między par-  
54 tonami siła oddziaływanego rośnie do nieskończoności, dlatego nigdy nie obserwuje się  
55 swobodnych cząstek obdarzonych ładunkiem kolorowym a jedynie związane w kolorowo  
56 obojętne hadrony.



Rysunek 2: Zależność silnej stałej sprzężenia od przekazu czteropędu. Źródło: [4].

## 57 1.2 Plazma kwarkowo-gluonowa

58 Odkrycie asymptotycznej swobody pozwoliło na sprawdzenie przewidywań *QCD* w warunkach  
59 bardzo wysokich temperatur oraz gęstości. Dla wystarczająco dużych gęstości hadrony zaczyna-  
60 ją na siebie zatracić, prowadząc do stworzenia stanu, w którym poszczególne hadrony przestają  
61 być odróżnialne [5]. Zasugerowane zostało także istnienie przejścia fazowego w temperaturze  
62 porównywalnej z masą pionów oraz w temperaturze niższej, ale przy odpowiednio dużych gę-  
63 stościach [6]. Stan materii powstały po osiągnięciu, któregoś z tych warunków nazywany jest  
64 plazmą kwarkowo-gluonową (ang. *quark-gluon plasma – QGP*). Obecnie przewiduje się, że  
65 materia w takim stanie istniała w pierwszych ułamkach sekund po Wielkim Wybuchu [7] oraz,  
66 że może się znajdować w jądrach gwiazd neutronowych [8].

67 Obecnie aby uzyskać dostęp do materii w stanie plazmy kwarkowo-gluonowej potrzebne są  
68 wysokoenergetyczne zderzenia cząstek. Powszechnie mówi się o niej w kontekście zderzeń cięż-  
69 kich jonów, chociaż istnieją także prace doszukujące się obecności *QGP* w mniejszych systemach  
70 np. w zderzeniach proton-proton [9], [10].

71 Cechą charakterystyczną *QGP* jest obecność wolnych kwarków i gluonów. Ze względu na  
72 uwiecznienie koloru w każdym innym stanie materii są one zawsze związane i tworzą hadrony.  
73 Wolne kwarki i gluony powstające w zderzeniach muszą zatem przejść przez proces hadroni-  
74 zacji, w którym rekombinują one ze spontanicznie wytworzonymi nowymi partonami, tworząc  
75 hadrony. W wyniku tego procesu, z każdego partonu obecnego w początkowym etapie zderzenia  
76 może powstać wiele cząstek poruszających się podobnym kierunkiem, tworząc stożek z wierzchoł-  
77 kiem blisko punktu interakcji wiązek. Taki stożek skolimowanych cząstek nazywany jest dżetem  
78 cząstek.

## 79 1.3 Dżety

80 Przedstawiona powyżej definicja dżetu nie jest precyzyjna z punktu widzenia pracy ekspery-  
81 mentalnej. W detektorze obserwuje się tylko cząstki w stanie końcowym, nie jest zatem możliwe  
82 przyporządkowanie cząstki według jej pochodzenia. W związku z tym, konieczne jest użycie algorytmu klasteryzującego, dostającego na wejściu tylko obserwowe eksperymentalnie cząst-  
83 ki. To jakie dżety zostaną zaobserwowane w danym zdarzeniu zależy od użytego algorytmu.  
84 Oznacza to, że precyzyjną definicję dżetu stanowi algorytm klasteryzujący wraz z zestawem  
85 parametrów. Obecnie najpowszechniej stosowanym algorytmem jest algorytm *anti-kt* [11].

86 Eksperimentalne ograniczenia związane z obserwacją tylko końcowego stanu oddziaływań  
87 nie występują w analizie danych z symulacji Monte Carlo (MC), gdzie ma się dostęp do peł-  
88 nej informacji na temat historii każdej cząstki. Nie należy jednak używać jej do klasteryzacji  
89 dżetów, gdyż utracona została cecha odpowiedniości między obiektymi nazywanymi dżetami  
90 w symulacji i w eksperymencie, która to cecha jest niewątpliwie jedną z podstawowych wyma-  
91 gań stawianych przed dobrą symulacją. Właściwym podejściem jest rekonstrukcja dżetów przy  
92 pomocy takiego samego algorytmu jak w przypadku danych eksperimentalnych.

93 Dżety wykorzystuje się w badaniach plazmy kwarkowo-gluonowej. Dają one pośredni wgląd  
94 we właściwości *QGP* na podstawie wpływu jaki wywiera na oddziałujące z nią partony. Przy-  
95 kładową obserwabłą mierzoną w zderzeniach ciężkich jonów jest czynnik modyfikacji jądrowej  
96  $R_{AA}$  (ang. *nuclear modification factor*), który jest miarą strat energii przez parton przechodzący  
97 przez medium. Jest to stosunek pędowych rozkładów dżetów cząstek zmierzonych w  
98 zderzeniach ciężkich jąder oraz w zderzeniach pp (przemnożonych przez liczbę binarnych zde-  
99 rzeń nukleon-nukleon przewidywanych przez model teoretyczny). Odchylenia od wartości 1 dla  
100 wysokich  $p_T$  są oznaką modyfikacji pędów dżetów przez gęste medium (w stosunku do zderzeń  
101 pp gdzie *QGP* nie powstaje), jest to tzw. tłumienie dżetów (ang. *jet quenching*). Rezultaty  
102 pomiarów  $R_{AA}$  w CMS: [12] i ALICE: [13].

103 Oprócz globalnego wpływu medium na dżety, analizuje się także różnice między dżetami  
104 pochodząymi z gluonów oraz kwarków o różnych zapachach (ang. *flavours*). Modele teore-  
105 tyczne przewidują między innymi większe straty energii w wyniku interakcji z *QGP* dla dżetów  
106 gluonowych niż kwarkowych [14] oraz zależność strat energii od masy partonu [15] – w tym  
107 przypadku precyzyjne pomiary rozróżniające typy dżetów pozwalają lepiej zrozumieć mecha-  
108 nizm odpowiadający za straty energii przez partony. Zagadnienie rozpoznania z jakiego rodza-  
109 ju partonu powstał dany dżet, nazywane jest identyfikacją lub tagowaniem dżetu. Ważną rolę  
110 w studiowaniu tego problemu odgrywają symulacje MC, które pozwalają określić wydajności  
111 poszczególnych technik tagowania dżetów na podstawie znajomości kanału produkcji każdej  
112 symulowanej cząstki.

## 114 1.4 Dżety *b*

### 115 1.4.1 Właściwości

116 Poza badaniami właściwości *QGP*, szczególne znaczenie ma identyfikacja dżetów pochodzących  
117 z ciężkich kwarków: *b* i *c*. Są one ważnym elementem w poszukiwaniu łamania symetrii *CP* w  
118 rozpadach hadronów B i D oraz innych sygnatur tzw. *Nowej Fizyki* wykraczającej poza ramy  
119 Modelu Standardowego. Kwarki *piękne* pojawiają się także często w kanałach rozpadu cząstek  
120 takich jak bozon Higgsa i kwark *t*.

121 Identyfikacja dżetów *b* jest sporym wyzwaniem ze względu na zdecydowanie częściej wy-  
122 stępujące dżety lekkie, tj. powstałe z hadronizacji kwarków *u,d,s* lub gluonów. Rozpoznawanie  
123 dżetów *b* bazuje na charakterystycznych właściwościach hadronów zawierających kwark piękny:  
124 relatywnie długim czasie życia oraz (w mniejszym stopniu) na ich pół-leptonowym rozpadach

125 o względnej częstotliwości rozpadu w tym kanale (ang. *branching ratio*) na poziomie 10%.

#### 126 1.4.2 Przegląd algorytmów używanych do identyfikacji dżetów $b$ na LHC

127 Używane w eksperymentach na LHC: ATLAS, CMS i ALICE algorytmy można podzielić na  
128 trzy kategorie: wykorzystujące wtórne wierzchołki, informację o odległości najbliższego zbliżenia  
129 (parametrze zderzenia) częstek tworzących dżet (ang. *Distance of Closest Approach – DCA*,  
130 *Impact Parameter – IP*) oraz identyfikujące produkty półleptonowych rozpadów pięknych lub  
131 powabnych hadronów. Dokładne opisy omawianych algorytmów można znaleźć w: [16], [17]  
132 (ATLAS), [18], [17] (CMS), [19], [20] (ALICE).

133 Najprostszym algorytmem jest dyskryminacja na podstawie istotności statystycznej (wynik  
134 pomiaru podzielony przez jego niepewność) odległości wtórnego wierzchołka od wierzchołka  
135 pierwotnego  $L$ . Jest to metoda wykorzystywana w każdym z trzech wymienionych eksperymentów (por. ATLAS: algorytm SV0, CMS: algorytm SSV, ALICE). Może być ona rozszerzona  
136 poprzez użycie dodatkowych zmiennych opisujących wtórny wierzchołek jak na przykład jego  
137 masa, ułamek niesionej przez niego całkowitej energii dżetu (por. ATLAS: SV1) lub użycie  
138 "pseudowierzchołków" (kombinacji dwóch częstek o dużych *DCA*) w celu poprawienia wydaj-  
139 ności detekcji o przypadki, w których wtórny wierzchołek nie został zrekonstruowany (por.  
140 CMS: CSV).

141 CMS: CSV).

142 Algorytmy wykorzystujące informację o poszczególnych częstekach mogą zasadniczo bazować  
143 albo na sumie logarytmów prawdopodobieństw pochodzenia każdej częstki z pierwotnego wierz-  
144 chołka (por. ATLAS: IP3D, CMS: JP) lub tym samym prawdopodobieństwie ale dla wybranej,  
145 np. drugiej lub trzeciej częstki na liście posortowanej według malejącego *IP* (por. ALICE i  
146 CMS: TC). Bardziej złożonym podejściem, w którym częstki nie są traktowane jako niezależne,  
147 jest użycie rekurencyjnych sieci neuronowych (por. ATLAS: RNNIP).

148 Do wykorzystania półleptonowego kanału rozpadu ciężkich hadronów do identyfikacji dżetów  
149  $b$  w eksperymentach ATLAS (SMT) i CMS (SE, SM) użyto wzmacnianych drzew decyzyjnych  
150 trenowanych na kilku ręcznie zdefiniowanych w tym celu zmiennych takich jak pęd leptonu  
151 transwersalny względem osi dżetu.

152 Znaczącą poprawę zdolności predykcyjnej można uzyskać łącząc kilka różnych modeli. Al-  
153 gorytm łączący może pobierać na wejściu albo tylko predykcje klasyfikatorów niższego poziomu  
154 (CMS: cMVAv2) lub dodatkowo także ich zmienne wejściowe (ATLAS: MV2, DL1). Do scalania  
155 używane są zwykle wzmacniane drzewa decyzyjne lub sieci neuronowe.

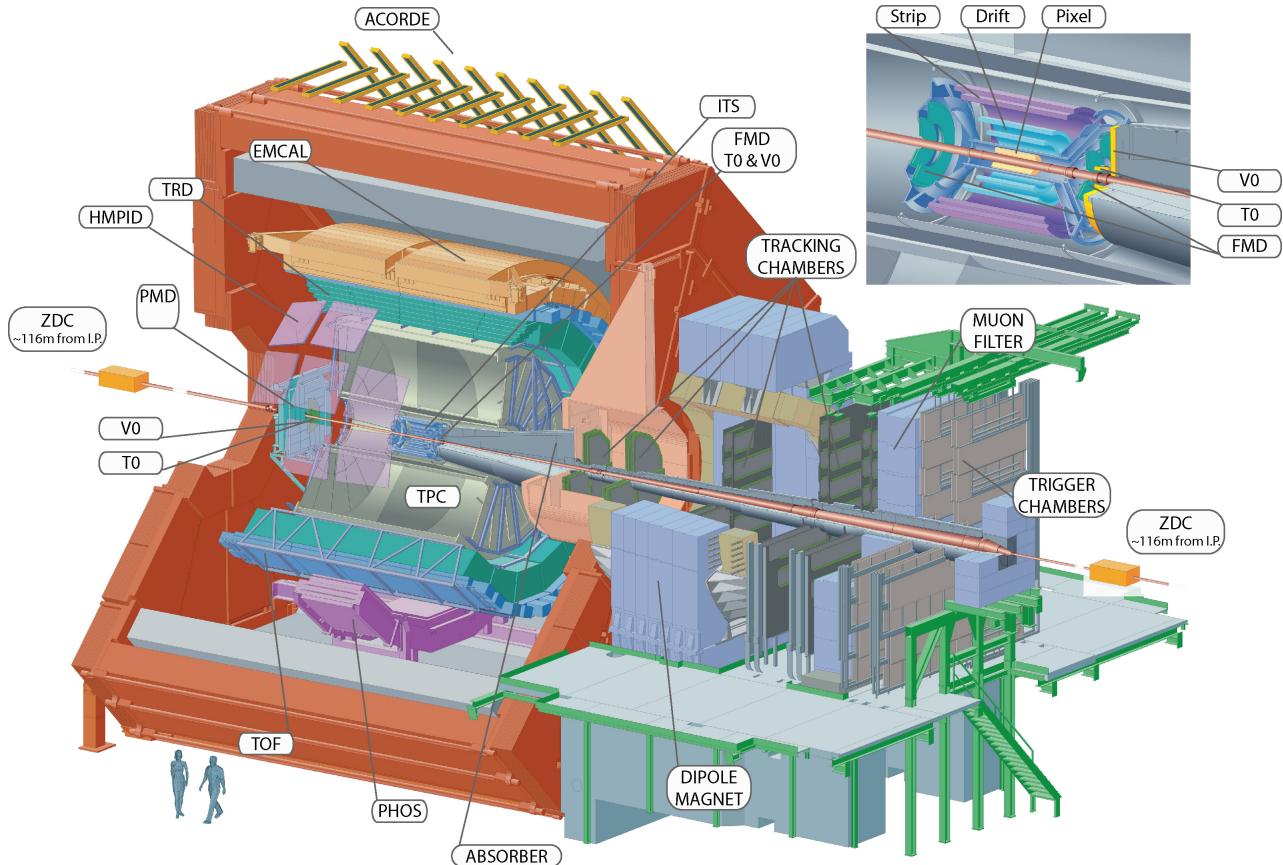
156 Przykład innego podejścia zaprezentowała współpraca przy eksperymencie LHCb, gdzie  
157 wykorzystano dwa zestawy wzmacnianych drzew decyzyjnych operujących na zmiennych zwią-  
158 zanych z wtórnymi wierzchołkami. Pierwszy zapewnia separację dżetów lekkich od ciężkich a  
159 drugi odróżnia dżety  $b$  od  $c$ . Do wyboru punktu pracy, zamiast jednowymiarowego rozkładu  
160 predykcji używany jest dwuwymiarowy rozkład wag przypisany przez oba klasyfikatory [21].

### 161 1.5 Eksperyment ALICE

162 Eksperyment ALICE [22], [23] jest jednym z czterech największych eksperymentów na LHC.  
163 Jest on dedykowany zderzeniom ciężkich jonów (w LHC są to jony ołowiu PbPb), ale mierzone  
164 są także mniejsze systemy, tj. proton-proton pp (głównie jako referencję dla pomiarów PbPb)  
165 oraz proton-ołów p+Pb, które dostarczają także okazji do badania asymetrycznych zderzeń.  
166 Cechą charakterystyczną pomiarów ciężkojonowych jest ich znacznie większa niż w przypadku  
167 zderzeń pp krotność, tj. liczba częstek wyprodukowana w pojedynczym zderzeniu. W przypadku  
168 zderzeń PbPb może powstawać nawet do 8000 naładowanych częstek na jednostkę pseudorapi-  
169 dity  $\eta$  (ang. *pseudorapidity*)<sup>1</sup>. Detektor ALICE został zoptymalizowany do mierzeniach takich

<sup>1</sup> $\eta = -\ln[\tan(\frac{\theta}{2})]$ , gdzie  $\theta$  jest kątem między wektorem pędu częstki a osią wiązki

170 przypadków, jak również pod kątem rekonstrukcji i identyfikacji cząstek o szerokim zakresie  
 171 pędów (100 MeV – 100 GeV).



Rysunek 3: Schemat detektora ALICE. Źródło: [24]

172 Detektor ALICE jest urządzeniem złożonym z wielu subdetektorów, schematycznie przed-  
 173stawionych na Rys. 3. Można je podzielić według pełnionej w pomiarach roli. ITS, TPC, TRD  
 174 oraz TOF pokrywają pełen kąt azymutalny oraz zakres pseudopospieszności  $|\eta| < 0.9$ .

- 175 • Detektory śladowe – mierzące trajektorie cząstek zakrzywiane w polu magnetycznym o  
 176 wartości  $B = 0.5$  T.
  - 177 – Inner Tracking System (ITS) – zespół krzemowych detektorów śladowych znajdują-  
 178 cy się najbliżej miejsca interakcji wiązek. Składa się on z 6 cylindrycznych warstw  
 179 o promieniach od 4 do 43 cm, wykonanych w trzech różnych technologiach. Jego  
 180 główną rolą jest rekonstrukcja pierwotnego oraz wtórnego wierzchołków. Bierze tak-  
 181 że udział w rekonstrukcji trajektorii i strat energetycznych cząstek, szczególnie tych  
 182 niskopędowych, które nie docierają do dalej położonych detektorów.
  - 183 – Time Projection Chamber (TPC) – dłuża na 5m i o takiej średnicy komora projekcji  
 184 czasowej. Jest to główny detektor śladowy ALICE, wraz z ITS służy do wyznacza-  
 185 nia trajektorii cząstek i na ich podstawie również wierzchołków zderzenia. Elektrony  
 186 uwolnione ze zjonizowanego przez poruszające się w nim naładowane cząstki ga-  
 187 zu dryfują wzdłuż kierunku wiązki w stronę końcowych elektrod. Następnie są tam  
 188 zbierane dostarczając informacji o dwóch współrzędnych toru cząstki: odległości od  
 189 wiązki i kącie azymutalnym. Trzecia składowa trajektorii jest otrzymywana na pod-  
 190 stawie czasu dotarcia elektronów do elektrod. TPC jest najwolniejszym detektorem  
 191 ALICE (ze względu na ograniczający czas dryfu elektronów wynoszący  $\sim 90 \mu\text{s}$ ),

192        użycie detektora tego typu podyktowane jest jego zdolnością do rozwikłania śladów  
193        tysięcy cząstek spodziewanych w centralnych zderzeniach PbPb.

194        Znajomość toru ruchu cząstki pozwala na wyznaczenie jej pędu. Oprócz dokładnej  
195        trajektorii każdej cząstki próbkowanej do 159 razy, TPC mierzy straty energii czą-  
196        stek  $dE/dx$ . Pozwala to na ich identyfikację na podstawie wzoru Bethego-Blocha,  
197        najwyższą zdolność rozdzielczą TPC osiąga dla cząstek o  $p_T < 1$  GeV.

198        • detektory służące identyfikacji cząstek (ang. *particle identification – PID*)

- 199        – Transition Radiation Detector (TRD) – detektor wykrywający promieniowanie przej-  
200        scia, służy głównie do odróżniania wysokopędowych ( $p_T > 1$  GeV) elektronów od  
201        pionów. Promieniowanie przejścia emitowane jest podczas przechodzenia relatywi-  
202        stycznych cząstek przez granicę ośrodków, jego intensywność jest proporcjonalna do  
203        czynnika Lorentza  $\gamma$ , co pozwala na odróżnienie cząstek o tym samym pędzie na pod-  
204        stawie różnicy mas (elektrony są ponad 250 razy lżejsze od pionów). TRD oprócz  
205        identyfikacji elektronów uczestniczy także w rekonstrukcji śladów wysokopędowych  
206        cząstek i może być użyty w systemach wyzwalania (ang. *trigger*).  
207        – Time-Of-Flight (TOF) – detektor czasu przelotu o zdolności rozdzielczej  $\sim 80$  ps.  
208        Pozwala na separację pionów i kaonów o pędach do ok. 2.5 GeV i protonów do 4  
209        GeV.  
210        – High-Momentum Particle Identification Detector (HMPID) – detektor typu RICH  
211        (ang. *ring-imaging Cherenkov*), wykrywający fotony emitowane podczas przejścia  
212        przez ośrodek naładowanej cząstki o prędkości większej od prędkości fazowej światła  
213        w tym ośrodku (promieniowanie Cherenkowa). Na podstawie kąta pod jakim emitowano  
214        są fotony określana jest prędkość cząstki. HMPID pozwala na identyfikację pio-  
215        nów, kaonów i protonów o  $p_T > 1$  GeV. Pokrywa przestrzeń kątów:  $1.2^\circ < \phi < 58.8^\circ$   
216        oraz  $|\eta| < 0.6$  (5% akceptancji TPC).

217        • kalorymetry

- 218        – Photon Spectrometer (PHOS) – elektromagnetyczny kalorymetr o wysokiej rozdziel-  
219        czości energetycznej i przestrzennej (podzielony na kryształy o rozmiarze poprzecznym  
220         $2.2 \times 2.2$  cm, co odpowiada rozmiarowi w dziedzinie  $\eta$ ,  $\phi$   $0.004 \times 0.004$ ). Pokrywa  
221        zakres pseudopospieszności  $|\eta| < 0.12$  i kąta azymutalnego równy  $100^\circ$ . PHOS ma  
222        za zadanie identyfikację i pomiar czteropędów fotonów, w szczególności tych niepo-  
223        chodzących z rozpadu innych cząstek (ang. *direct photons*) oraz lekkich mezonów  
224        neutralnych (np.  $\pi^0$ ) przez dwufotonowy kanał rozpadu.  
225        – Electromagnetic Calorimeter (EMCal) – drugi elektromagnetyczny kalorymetr ALI-  
226        CE o mniejszej ziarnistości ( $\Delta\eta, \Delta\phi = 0.014 \times 0.014$ ), ale dużo większej akceptancji  
227        ( $|\eta| < 0.7$ ,  $\Delta\phi = 107^\circ$ ). EMCal poprawia możliwości ALICE w zakresie pomiarów  
228        tłumienia dżetów, pozwalając na wyznaczanie neutralnej składowej energii dżetów  
229        (energii niesionej przez neutralne cząstki). Dzięki innej charakterystyce dla elektro-  
230        nów i hadronów (elektrony typowo deponują niemal całą energię a hadrony tylko  
231        niewielką część) pozwala je odróżnić na podstawie stosunku zmierzonej w nim ener-  
232        gii do wyznaczonego wcześniej pędu  $E/p$ . EMCal może być użyty także w szybkim  
233        systemie wyzwalania, do selekcji przypadków z dżetami oraz wysokoenergetycznymi  
234        fotonami i elektronami.  
235        • Muon spectrometer – spektrometr mionowy, złożony z dwóch pasywnych absorberów,  
236        znajdujących się między nimi 10 warstw detektora śladowego oraz komór systemu wy-  
237        zwalającego na końcu. Przedni absorber, gruby na 4 metry ( $\sim 60X_0$ ) wykonany z betonu

i grafitu, zatrzymuje hadrony oraz miony o niższych energiach (np. z rozpadów pionów i kaonów). Jest on zoptymalizowany aby minimalizować rozpraszanie mionów i zapewnić ochronę pozostałych detektorów ALICE przed wtórnymi cząstkami powstały w jego materiale. Komory pozycjoczułe mają zdolność rozdzielczą ok.  $100 \mu\text{m}$ , co pozwala osiągnąć wysoką rozdzielczość przy wyznaczaniu masy niezmienniczej rzędu  $100 \text{ MeV}/c^2$ . Spektrometr mionowy służy głównie do mierzenia mezonów wektorowych ( $\omega$ ,  $\phi$ ,  $J/\Psi$ ,  $\Upsilon$ ) rozpadających się w kanale  $\mu^+ \mu^-$ .

- Detektory przednie, wyznaczające min. centralność zderzeń oraz płaszczyznę reakcji.

- ZDC – zespół czterech kalorymetrów (po dwa do pomiaru protonów i neutronów, gdyż ich tory są rozdzielane przez pole magnetyczne) mierzących energię nukleonów nieuczestniczących w zderzeniu tzw. obserwatorów, co pozwala na określenie liczby nukleonów oddziałujących, tzw. uczestników. Znajdują się one 116 m od miejsca interakcji.
- PMD – detektor mierzący krotności oraz rozkład przestrzenny fotonów
- FMD – krzemowy detektor paskowy mierzący precyjnie liczbę naładowanych cząstek w zakresie pseudopospieszności wykraczającym poza akceptancję detektora ITS.
- V0 – liczniki scyntylacyjne położone po obu stronach detektora, używane w systemie wyzwalania o minimalnym obciążeniu (ang. *minimum bias trigger*) – wymóg obecności sygnału w obu detektorach pozwala na odrzucenie przypadków tła z oddziaływania wiązki protonów z reszkami gazu obecnymi w rurach próżniowych.
- T0 – dostarcza dokładny czas interakcji potrzebny dla detektora TOF, pozwala także na śledzenie świetlności w czasie rzeczywistym.

## 260 2 Uczenie maszynowe

261 Uczenie maszynowe jest bardzo szerokim i obecnie dynamicznie się rozwijającym obszarem  
262 nauki. Występuje w wielu odmianach łącząc w sobie w zależności od wariantu wiele dziedzin  
263 takich jak matematyka (statystyka, algebra) informatyka (algorytmika, teoria informacji) a  
264 także elementy robotyki i sterowania. Dziedzinami, w których jest najczęściej wykorzystywane  
265 są min. widzenie maszynowe, przetwarzanie języka naturalnego, autonomiczne roboty i pojazdy,  
266 systemy decyzyjno - eksperckie, optymalizacyjne oraz rekomendacyjne.

267 W tej pracy wykorzystywana jest gałąź uczenia maszynowego nazywana uczeniem nadzorowanym lub "uczeniem z nauczycielem" (ang. *supervised learning*), gdzie uczenie występuje na podstawie poprawnie oznaczonych przykładów. Terminami bliskoznacznymi dla tak rozumianego uczenia maszynowego są uczenie statystyczne (ang. *statistical learning*) i rozpoznawanie wzorców (ang. *patterm recognition*).

272 Problem identyfikacji dżetów jest klasycznym przykładem zagadnienia klasyfikacji, gdzie  
273 poprawna odpowiedź jest jedną ze skończonej ilości opcji (klas) w przeciwieństwie do regresji,  
274 gdzie szukana odpowiedź algorytmu ma charakter ciągły.

275 Występuje wiele algorytmów uczenia maszynowego takich jak regresja liniowa i logistyczna,  
276 drzewa decyzyjne i ich wariacje, maszyny wektorów wspierających, sztuczne sieci neuronowe oraz wiele innych [25], [26]. Uczenie polega na znalezieniu pewnej funkcji dopasowującej  
277 do przyjmowanego na wejściu zestawu (wektora) cech (zmiennych, kolumn) pewną odpowiedź  
278 (predykcję), która minimalizuje zadaną funkcję straty. Jej rolę w przypadku regresji często  
279 pełni błąd średniokwadratowy a w przypadku klasyfikacji np. entropia krzyżowa (ang. *cross  
280 entropy*)<sup>2</sup>. Różne algorytmy szukają przy tym funkcji dopasowującej należącej do różnych klas  
281 funkcji: przykładowo klasyczne drzewa decyzyjne przeszukują tylko przestrzeń funkcji dających  
282 się opisać skończonym zbiorem reguł "jeśli – to" (ang. *if – else*).

284 W pracy wykorzystane zostały dwa rodzaje algorytmów: wzmacniane drzewa decyzyjne oraz  
285 sieci neuronowe.

### 286 2.1 Wzmacniane drzewa decyzyjne

287 Wzmacniane drzewa decyzyjne są jednym z rozwinięć klasycznego algorytmu drzewa decyzyjnego. Pojedyncze drzewo decyzyjne dzieli przestrzeń cech uczących przy pomocy prostopadłych  
288 cięć, na mniejsze/większe niż zadana wartość w przypadku zmiennej ciągłej lub na należące/nie  
289 należące do danej klasy w przypadku zmiennej kategorycznej. Każdy podział, nazywany węzłem, daje dwie gałęzie, które można dalej niezależnie dzielić aż do ostatniego poziomu (liści). Kolejne podziały wybierane są tak, aby zbiory przykładów wpadające do poszczególnych gałęzi  
290 były jak najbardziej jednorodne. Stosuje się różne miary nieporządku takie tak: indeks Gini  $G_L$   
291 lub entropia  $S_L$ <sup>3</sup>.

295 Drzewa decyzyjne są często łączone w komitety klasyfikatorów (ang. *ensemble methods*). Wiele "słabych" klasyfikatorów jest łączonych w jeden "silny" na dwa sposoby: workowanie  
296 (ang. *bagging*) [27] oraz wzmacniania (ang. *boosting*) [28], które są często ze sobą porównywane.

298 *Bagging* – w zastosowaniu dla drzew decyzyjnych nazywany algorytmem lasów losowych  
299 (ang. *random forest*) - polega na wytrenowaniu wielu drzew, każdego na podstawie  $N$  przykładów losowo wylosowanych z powtórzeniami spośród  $N$ -licznego zbioru treningowego. Dodatko-

<sup>2</sup>  $J = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$ , gdzie  $y_i$  to prawidłowa klasa  $i$ -tego przykładu a  $\hat{y}_i$  to predykcja algorytmu

<sup>3</sup>  $G_L = 1 - \sum_k p_k^2$  oraz  $S_L = \sum_k -p_k \log p_k$ , gdzie  $p_k$  to stosunek liczby przypadków klasy  $k$  do liczby wszystkich przypadków w liściu  $L$

301 wo, do uczenia każdego drzewa używa się tylko podzbioru wszystkich cech uczących. Końcową  
302 predykcję algorytmu otrzymuje się poprzez "głosowanie" wszystkich drzew z odpowiednimi  
303 wagami.

304 *Boosting* – wzmacniane drzewa decyzyjne (ang. *boosted decision trees*) – jest metodą po-  
305 dobną do *baggingu*. Główną różnicą jest zwiększanie wag przykładom uczącym, które przez po-  
306 przednie drzewo zostały źle zaklasyfikowane – każde kolejne drzewo koncentruje się bardziej na  
307 poprawie błędów poprzednich drzew. Widać tu kolejną ważną cechę odróżniającą obie metody:  
308 *boosting* jest algorytmem sekwencyjnym podczas gdy *bagging* daje się trywialnie zrównoleglić  
309 (każde drzewo trenowane jest w osobnym wątku).

### 310 **Parametry i sposób trenowania drzew decyzyjnych na analizowanych danych**

311 W niniejszej pracy wykorzystano wzmacniane drzewa decyzyjne zaimplementowane w wydaj-  
312 nej bibliotece **XGBoost** [29]. Szybkość obliczeń jest bardzo ważna, gdyż oprócz komfortu pracy  
313 z algorytmem, przekłada się na jakość otrzymanych wyników – krótszy czas obliczeń oznacza  
314 możliwość przeprowadzenia większej ilości eksperymentów i lepsze dobranie parametrów oraz  
315 danych. Implementacja wzmacnianych drzew decyzyjnych w **XGBoost** wykorzystuje wszystkie  
316 rdzenie procesora, pomimo że sam algorytm ma charakter sekwencyjny – jest to możliwe dzięki  
317 paralelizacji procesu tworzenia każdego drzewa (przed każdym podziałem konieczne jest spraw-  
318 dzenie pewnej ilości możliwych zmiennych i wartości progowych i ten proces jest wykonywany  
319 równolegle).

320 Dzięki szybkiemu uczeniu się algorytmu, możliwe było użycie kosztownego obliczeniowo au-  
321 tomatycznego przeszukiwania przestrzeni parametrów przy pomocy przeszukiwania losowego  
322 (ang. *random search*), które jest zwykle preferowane nad przeszukiwanie sieciowe [30]. W tym  
323 celu cały zbiór danych dzielony był na dwie części: trenującą oraz testową (80/20%). Następ-  
324 nie algorytm był trenowany i oceniany z użyciem trzy- lub pięciokrotnej walidacji krzyżowej  
325 (ang. *cross-validation*) na zbiorze trenującym dla różnych zestawów parametrów. Model z naj-  
326 lepszym wynikiem uzyskanym w walidacji krzyżowej był sprawdzany na zbiorze testowym.

327 Parametry optymalizowane w opisanym procesie to:

- 328 • *max\_depth* – maksymalna głębokość każdego drzewa (niekoniecznie osiągana)
- 329 • *n\_estimators* – liczba drzew
- 330 • *learning\_rate* – parametr szybkości uczenia, komplementarny do *n\_estimators*, w praktyce  
331 można ustalić liczbę drzew i szukać optymalnej szybkości uczenia
- 332 • *subsample*, *colsample\_bytree*, *colsample\_bylevel* – parametry regularyzacyjne określające  
333 ułamek kolejno: danych użytych do trenowania każdego drzewa, kolumn użytych w ka-  
334 dym drzewie (cechy losowane raz dla danego drzewa), kolumn użytych przy każdym po-  
335 dziale (cechy losowane przy każdym podziale)
- 336 •  $\gamma$  – minimalny zysk w postaci zmniejszenia wartości funkcji straty konieczny do wykonania  
337 podziału

## 338 **2.2 Sieci neuronowe**

339 Sieci neuronowe (ang. *neural networks* – *NN*) są szczególnym algorytmem uczenia maszyno-  
340 wego. Występują w bardzo wielu odmianach i są wykorzystywane w rozwiązywaniu szerokiej  
341 gamy problemów. Nawet bardzo pobieżny opis sieci neuronowych wymaga dużo więcej miejsca  
342 niż może być temu poświęcone w tej pracy. Wprowadzenia do sieci neuronowych od podstaw  
343 można znaleźć m.in. w [31] lub [32]. Tu przedstawione zostaną wyłącznie wybrane zagadnienia

344 mające ścisłejšzy związek z pracą. Używane mogą być terminy, których znaczenie wyjaśniane  
345 jest w podanych źródłach.

346 W niniejszej pracy, wykorzystane zostały dwa rodzaje sieci neuronowych: sieci w pełni połącz-  
347 czone (ang. *fully connected NN – FC NN*), nazywane także wielowarstwowymi perceptronami  
348 (ang. *multi-layer perceptron – MLP*) oraz sieci konwolucyjne (ang. *convolutional NN – Co-*  
349 *nvNets, CNN*).

### 350 Sieci w pełni połączone

351 W nierekurencyjnych sieciach neuronowych (tylko takie są używane w tej pracy), informacja  
352 jest przekazywana kolejno od warstw wejściowych, poprzez warstwy ukryte aż do wyjściowej. W  
353 sieciach typu *FC* wszystkie warstwy składają się z identycznych neuronów – każdy neuron do-  
354 staje na wejściu wektor, natomiast zwraca skalar – wartość pewnej zadanej, nieliniowej funkcji,  
355 jako argument podając średnią ważoną z elementów wektora wejściowego. Wartości zwracane  
356 przez neurony w danej warstwie składają się na wektor wejściowy dla neuronów kolejnej  
357 warstwy. Wejściem dla pierwszej warstwy są natomiast kolejne przykłady ze zbioru uczącego.  
358 Trenowanie sieci neuronowych polega na zmienianiu wag (parametrów) w liczonej w każdym  
359 neuronie średniej, każdy neuron posiada własny, niezależny zestaw wag.

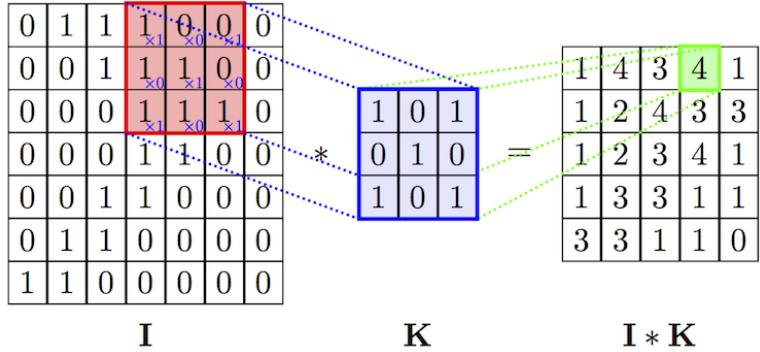
360 Istnieje twierdzenie o sieciach neuronowych jako uniwersalnych aproksymatorach funkcji  
361 (ang. *universal approximation theorem*) [33], mówiące, że już sieć neuronowa o jednej warstwie  
362 ukrytej jest zdolna do przybliżenia dowolnej funkcji z dowolną dokładnością. Twierdzenie to  
363 nie podaje niestety liczby potrzebnych neuronów a przede wszystkim – sposobu ich trenowania.  
364 Trenowanie jest prostsze w przypadku zastosowania wielu warstw, które odpowiadają kolejnym  
365 poziomu abstrakcji jednak nadal jest dużym wyzwaniem ze względu na fakt, że nawet stosun-  
366 kowo niewielka sieć może posiadać bardzo dużą liczbę parametrów, przykładowo sieć o czterech  
367 warstwach, w każdej po 128 neuronów ma ich ponad 65 tysięcy.

### 368 Sieci konwolucyjne

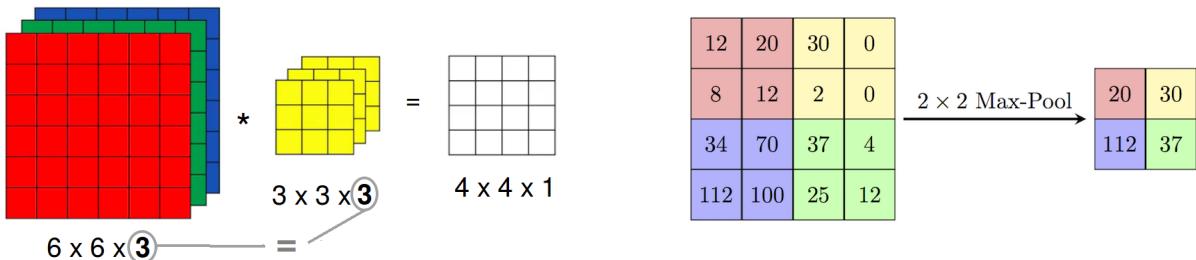
369 Jednym ze sposobów na ograniczenie liczby trenowanych parametrów jest użycie konwolucyj-  
370 nych sieci neuronowych [34] (bardziej poprawną choć rzadko używaną nazwą w języku polskim  
371 jest sieć splotowa). Są one inspirowane połączeniami w korze wzrokowej zwierząt i wywodzą  
372 się z badań w obszarze widzenia komputerowego, gdzie liczby parametrów są szczególnie duże  
373 (wektor wejściowy ma wymiar równy liczbie pikseli w obrazie), na takim przykładzie również  
374 najłatwiej zrozumieć ich działanie.

375 Sieci konwolucyjne różnią się od sieci typu *FC* tym, że część wag połączeń między warstwa-  
376 mi jest dzielona. Występuje w nich nowy rodzaj warstwy, nazywany warstwą konwolucyjną.  
377 Każda jednostka w warstwie konwolucyjnej (filtr) ma pewną stałą (niewielką) liczbę wag. Po-  
378 łączanie z dużym wejściem realizowane jest przez powielanie tych samych wag w połączeniach  
379 z kolejnymi fragmentami wektora wejściowego (por. Rys. 4). Rezultatem działania filtra na  
380 macierz jest wynik operacji splotu. Liczba parametrów przypadająca na każdy filtr jest równa  
381 jego rozmiarowi i nie zależy od wielkości wektora wejściowego.

382 W przypadku gdy zamiast wejścia dwuwymiarowego (jak np. obraz czarno-biały), mamy do  
383 czynienia z wejściem trójwymiarowym (np. trzeci wymiar to kolejne kolory w kodowaniu RGB),  
384 filtry również muszą mieć trzy wymiary, przy czym rozmiar w ostatnim wymiarze musi być  
385 równy rozmiarowi w tym kierunku wektora wejściowego. Wynik operacji splotu jest ponownie  
386 dwuwymiarowy, gdyż filtr przesuwany jest tylko w dwóch pierwszych wymiarach. Trzeci wymiar  
387 powstaje przez składanie kolejnych filtrów. Widać zatem, że również w przypadku gdy na  
388 wejścia podawany jest obraz czarno-biały, filtry w kolejnych warstwach konwolucyjnych (oprócz  
389 pierwszej) mają po trzy wymiary.



Rysunek 4: Schemat działania pojedynczego filtra z warstwy konwolucyjnej (operacja splotu).  
Źródło: [35].



Rysunek 5: Działanie pojedynczego filtra (3D) na wejście o trzech wymiarach.  
Źródło: [36].

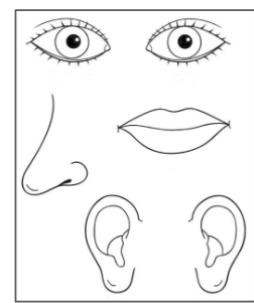
Rysunek 6: Działanie warstwy typu *max-pool*. Źródło: [35].

oprócz warstw konwolucyjnych, w sieciach tego typu stosowane są także tzw. warstwy typu *max-pooling*. Zasada jej działania jest bardzo prosta: wykonuje funkcję *maksimum* na zadanym fragmencie obrazu (por. Rys. 6). Ich rolą jest zmniejszanie rozmiaru przekazywanej w sieci informacji.

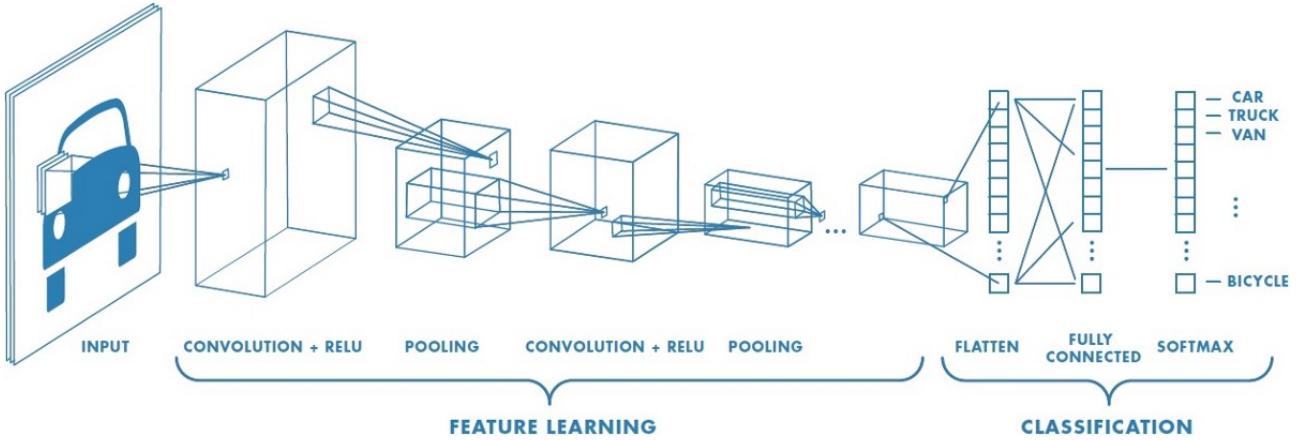
Typowa architektura stosowana w przypadku sieci konwolucyjnych jest następująca: najpierw warstwy konwolucyjne (pomiędzy nimi czasem warstwy typu *max-pool*), następnie wszystkie filtry są rozwijane i składane w długi jednowymiarowy wektor, który przekazywany jest do warstw typu *FC*. W przypadku problemu klasyfikacji, na końcu znajduje się jeszcze warstwa typu *softmax* normalizująca wyjście z sieci do jedynki (por. Rys. 7).

Sieci konwolucyjne posiadają dwie właściwości odróżniające je od *MLP*:

- niezmienniczość względem przesunięcia (ang. *translation invariance*) – głównie za sprawą dzielenia wag oraz obecności warstw typu *max-pool*, położenie danej cechy na obrazie jest niemal bez znaczenia (obraz po prawej stronie byłby rozpoznany jako twarz)
- lokalność połączeń – filtry obejmują tylko kilka sąsiednich pikseli (tam zwykle występują najsilniejsze zależności) nie są w stanie dostrzec cechy rozciągniętej na obszar większy od rozmiaru filtra



Rysunek 8: Źródło: [38].

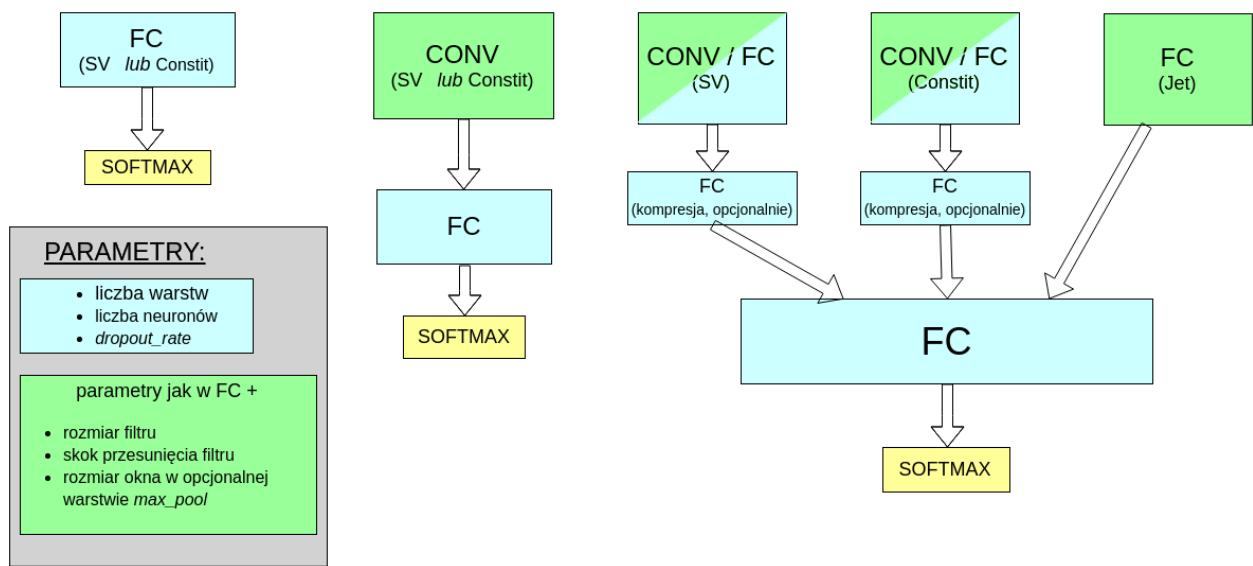


Rysunek 7: Typowa struktura stosowana w sieciach konwolucyjnych. Warstwy konwolucyjne mają za zadanie wydobywać cechy, na podstawie których późniejsze warstwy dokonują klasyfikacji. Widoczna jest charakterystyczna stopniowa zmiana rozmiaru przekazywanej macierzy: rozmiar poprzeczny maleje kosztem głębokości, co odpowiada rosnącej liczbie filtrów i malejącym rozmiarowi obszaru po jakim są one przesuwane. Źródło: [37].

#### 410 Hiperparametry i trenowanie sieci neuronowych na analizowanych danych

411 Parametry sieci, których wartości są określane przez projektanta sieci, takie jak liczba warstw  
412 ukrytych są nazywane hiperparametrami (dla odróżnienia od parametrów - wag połączeń).

413 Testowane były trzy architektury: sieci w pełni połączone, sieci konwolucyjne oraz sieć  
414 złożona z dwóch gałęzi, osobnych dla wtórnych wierzchołków i częstek tworzących dżet (por.  
415 Rozdz. 3) przedstawione schematycznie na Rys. 9. Do trenowania sieci wykorzystano wysoko-  
416 poziomową bibliotekę Keras [39] korzystającą z silnika obliczeniowego zaimplementowanego w  
417 TensorFlow [40].



Rysunek 9: Schematyczne przedstawienie trzech testowanych rodzin architektur sieci. Każdy blok odpowiada kilku warstwom danego typu. Bloki warstw typu *FC* i opisane jako "kompresja" składały się z kilku neuronów i miały za zadanie zredukować całą informację z danej gałęzi do wektora kilku liczb.

418 Zestaw hiperparametrów definiujący działanie sieci w pełni połączonej:

- 419 • liczba warstw ukrytych
- 420 • liczba neuronów w każdej warstwie
- 421 • funkcja aktywacji – nielinowa funkcja aplikowana przed zwróceniem wartości w każdym  
422 neuronie, najpopularniejsze to *tanh*, *ReLU* ( $f(x) = \max(0, x)$ ) oraz funkcja sigmoidalna  
423 ( $f(x) = \frac{1}{1+\exp(-x)}$ )
- 424 • algorytm optymalizacyjny – spadek gradientowy lub jego wariacje
- 425 • parametr szybkości uczenia i jego modyfikacje w trakcie uczenia
- 426 • liczba przykładów trenujących przetwarzanych w jednym kroku uczenia (ang. *batch\_size*)  
427 – im większy tym szybsze jest trenowanie sieci (dzięki wydajnym operacjom macierzowym), natomiast może się to odbywać kosztem precyzji
- 428
- 429 • liczba epok uczenia – ile razy będzie pokazywany sieci każdy przykład
- 430 • opcjonalnie: warunki stopu (ang. *early stopping*) w razie osiągnięcia *plateau* (wysycenia  
431 procesu uczenia)
- 432 • opcjonalnie: regularyzacja przy pomocy różnych technik (zwykle konieczna)

433 Ponadto dla sieci konwolucyjnych:

- 434 • liczba warstw konwolucyjnych i liczba filtrów w każdej warstwie
- 435 • obecność lub brak warstw *max-pool* i rozmiar ich okna
- 436 • rozmiar filtrów i długość skoku przy ich przesuwaniu

437 Same dwie pierwsze wielkości dają nieograniczoną liczbę konfiguracji. Czas trenowania sieci  
438 neuronowych jest rzędu wielkości większej niż drzew decyzyjnych, dlatego przyjęto szereg kro-  
439 ków mających na celu zmniejszenie przeszukiwanej przestrzeni hiperparametrów. Na podstawie  
440 wstępnych testów oraz różnych wskazówek dostępnych w literaturze przyjęto:

- 441 • *batch\_size* zawsze równy 64 (inne testowane wartości: 16, 32, 128)
- 442 • za algorytm optymalizacyjny przyjęto algorytm o nazwie *Nadam* [41], tj. rozwinięcie al-  
443 gorytmu *Adam* [42] o parametr Nesterova (inne testowane to zwykły spadek gradientowy  
444 oraz *Adam*)
- 445 • funkcję aktywacji: *ReLU*
- 446 • liczba epok równa 50, 100 lub 200, zrezygnowano z *early stopping*
- 447 • stałe w trakcie treningu wartości parametru szybkości uczenia
- 448 • spośród technik regularyzacyjnych testowano wyłącznie *dropout* [43] z prawdopodobień-  
449 stwem odrzucenia równym 0.1, 0.2 lub 0.5
- 450 • kilka wybranych kombinacji dla zestawu parametrów: rozmiar filtra, długość skoku i roz-  
451 miar okna w warstwach *max-pool* – takie same w kolejnych warstwach

- 452     ● liczby neuronów/filtrów w warstwach będące zawsze potęgami dwójki oraz stałą liczbę w  
453        kolejnych warstwach lub zmieniającą się o stały czynnik, np. 256-128-64, 128-128-128 lub  
454        16-32-64
- 455     ● liczba warstw *FC*: 2-8, konwolucyjnych 2-6

456     Nawet po przyjęciu powyższych uproszczeń nie sposób sprawdzić wszystkich możliwych  
457        zestawów hiperparametrów, dlatego sposób ich dobierania w kolejnych testach był mocno em-  
458        piryczny. Dostępne dane dzielone były na trzy zbiory: trenujący, walidacyjny i testowy. Wobec  
459        braku warunków stopu, zbiór walidacyjny użyty był wyłącznie do porównywania różnych ze-  
460        stawów parametrów, tak aby wynik testowy pozostał nieobciążony.

461     Zgodnie z zasadą ortogonalizacji działań, proces doboru hiperparametrów dzielono na dwie  
462        części: najpierw starano się uzyskać jak najlepsze wyniki na zbiorze uczącym, a dopiero później  
463        zmusić algorytm do lepszej generalizacji na zbiorze testowym przez zwiększoną regularyzację i  
464        modyfikację parametru szybkości uczenia.

## 465     **2.3 Dyskusja użycia dwóch algorytmów**

466     Użycie więcej niż jednego algorytmu ma wiele zalet. Po pierwsze daje możliwość porównania  
467        wyników. Pozwala to na oszacowanie błędu *Bayesowskiego* (najniższego możliwego do osiągnię-  
468        cia przez jakikolwiek algorytm błędu). Jest to bardzo ważne w sytuacji, gdy nie dysponuje  
469        się innym oszacowaniem tego błędu (w wielu problemach naturalnych dla człowieka jak roz-  
470        poznawanie obiektów na obrazkach jest nim błąd ludzki lub też błąd popełniany przez zespół  
471        ekspertów w bardziej zaawansowanych zastosowaniach).

472     Po drugie, wykorzystane zostały dwa algorytmy mocno różniące się w swojej naturze, co po-  
473        zwala wykorzystać cechy każdego z nich w analizie: przykładowo sieci neuronowe dobrze radzą  
474        sobie z nieustrukturyzowanymi danymi – potrafią tworzyć wysoko poziomowe cechy na podsta-  
475        wie niskopoziomowego wejścia (np. położenia oka na zdjęciu twarzy na podstawie pixeli). Są  
476        natomiast trudne w interpretacji i często traktowane są jako tzw. "czarne skrzynki" (ang. *black*  
477        *box*). Oprócz tego, liczba możliwych konfiguracji sieci jest ogromna i przez to niemożliwe jest  
478        stwierdzenie czy wykorzystane zostały pełne ich możliwości.

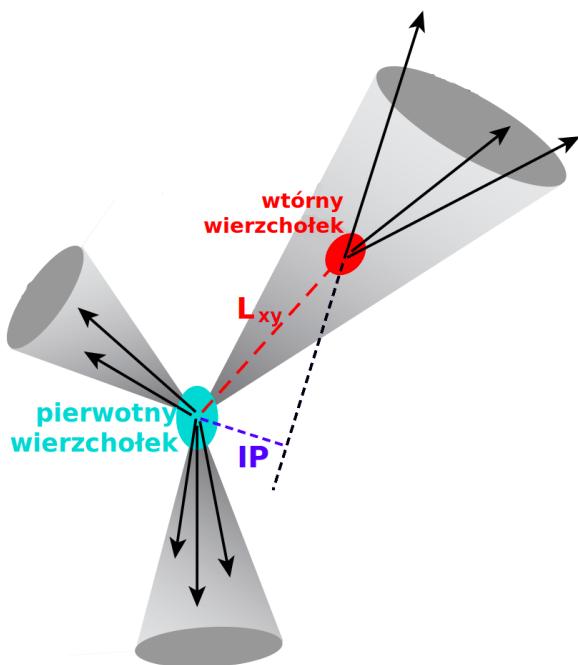
479     Z kolei drzewa decyzyjne posiadają stosunkowo niewielką liczbę parametrów, a ich trenowa-  
480        nie jest bardzo szybkie co pozwala na ich ekstensywne przeszukiwanie i otrzymanie wyników,  
481        które można uznać za optymalne dla tego algorytmu. Ponadto, w przypadku drzew istnieją  
482        niewymagające dodatkowych obliczeń miary użyteczności poszczególnych zmiennych, co daje  
483        wgłębne w działanie algorytmu i poprawia intuicyjne zrozumienie jego predykcji.

### 484 3 Dane

485 Dane użyte w analizie pochodzą z symulacji Monte Carlo zderzeń proton-proton przy energii  
486 w układzie środka masy równej  $\sqrt{s} = 13$  TeV dostępnych na serwerach eksperymentu ALICE.  
487 Są to pełne symulacje detektora ALICE, wykorzystujące generator zderzeń Pythia8 [44] (*tune:*  
488 *Pythia8Jets\_Monash2013* [45]) oraz pakiet Geant3 [46] do transportu cząstek przez materiał  
489 detektora.

490 Do rekonstrukcji dżetów wykorzystany został algorytm *anti-kt* z parametrem  $R = 0.4$  zaim-  
491 plementowany w pakiecie FASTJET [47]. Dżetów poszukiwano wyłącznie wśród cząstek naładowa-  
492 nych (ang. *charged jets*) ze względu na słabe pokrycie przestrzeni fazowej przez kalorymetry  
493 w eksperymencie ALICE.

494 Do analizy wybrano dżety o  $p_T$  większym niż 15 GeV i mieszczące się w całości w akcep-  
495 tancji detektora *TPC*, tj.  $|\eta| < 0.9$ , co przy użytym parametrze rozmiaru dżetu  $R = 0.4$ , daje  
496 ograniczenie na pseudopospieszność  $|\eta| < 0.5$  dla osi dżetu.



Rysunek 10: Rysunek ilustrujący znaczenie używanych wielkości:  $L_{xy}$  oraz parametru zderzenia  $IP$ . Źródło: [48].

497 Dla każdego dżetu obliczony został szereg wielkości, które można podzielić na zmienne  
498 na poziomie dżetu, związane z wtórnymi wierzchołkami oraz cząstkami tworzącymi dżet. Za  
499 potencjalne wtórne wierzchołki uznaje się wszystkie kombinacje trzech cząstek spełniających  
500 pewne dosyć luźne kryteria jak  $p_T > 0.15$  GeV (rozważane są wyłącznie trzy-cząstkowe wtórne  
501 wierzchołki), stąd ich liczba może być dużo większa od liczby cząstek tworzących dżet.

502 Lista używanych zmiennych:

- 503 • Zmienne na poziomie dżetu:
  - 504 –  $\eta, \phi$  – pseudopospieszność i kąt azymutalny osi dżetu
  - 505 –  $p_T$  – pęd poprzeczny dżetu
  - 506 – masa dżetu [#REF]

- 507        – powierzchnia dżetu – liczona w płaszczyźnie ( $\eta, \phi$ ), do powierzchni dżetu zaliczany  
 508        jest element w powierzchni w którym dodanie części o nieskończonym małym pędzie  
 509        poprzecznym sprawi, że zostanie ona zaliczona do tego dżetu [49]  
 510        – gęstość tła (w danym zdarzeniu)  
 511        –  $N_{SV}$  – liczba wtórych wierzchołków  
 512        –  $N_{Constit}$  – liczba częstek tworzących dżet

513     • Zmienne opisujące cząstki tworzące dżet:

- 514        –  $\eta, \phi$  – pseudopospieszność i kąt azymutalny cząstki względem osi dżetu  
 515        –  $p_T$  – pęd poprzeczny cząstki  
 516        –  $IP_D$  – rzut na kierunek poprzeczny wektora parametru zderzenia  
 517        –  $IP_Z$  – rzut na oś  $z$  wektora parametru zderzenia

518     • Zmienne opisujące wtórne wierzchołki:

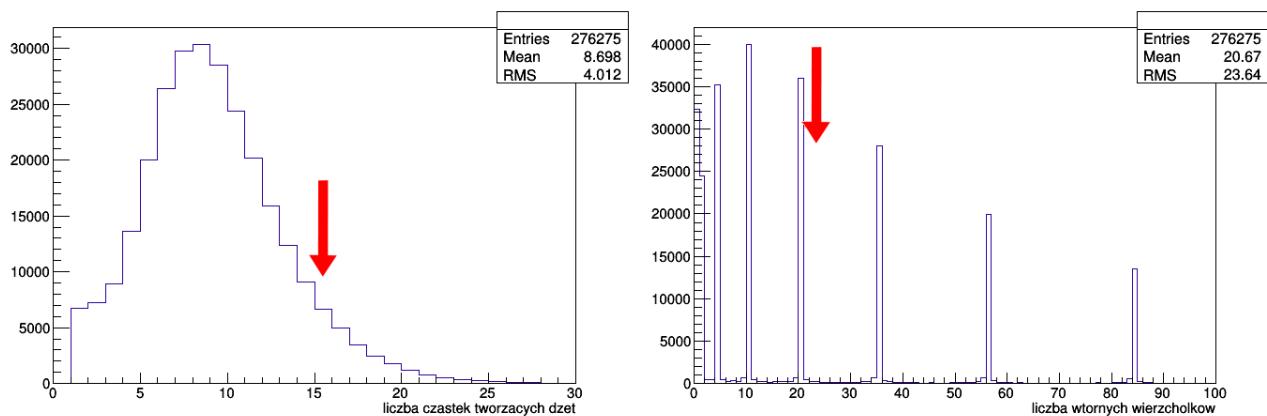
- 519        –  $L_{xy}$  – odległość między pierwotnym a wtórnym wierzchołkiem (ang. *decay length*)  
 520        –  $\sigma_{L_{xy}}$  – niepewność wyznaczenia  $L_{xy}$   
 521        –  $\sigma_{vertex} = \sqrt{d_1^2 + d_2^2 + d_3^2}$  – rozrzut śladów (ang. *tracks*) wokół wtórnego wierzchołka,  
 522        gdzie  $d_i$  to odległość najbliższego zbliżenia śladu / odległość najbliższego przelotu  
 523        do wtórnego wierzchołka (ang. *distance of closest approach – DCA*)  
 524        –  $M_{inv} = \sqrt{(E_1 + E_2 + E_3)^2 - (\vec{p}_1 + \vec{p}_2 + \vec{p}_3)^2}$  – masa niezmiennica wierzchołka, gdzie  
 525         $E_i, p_i$  to energia i pęd  $i$ -tej cząstki tworzącej wierzchołek  
 526        –  $\chi^2/Ndf$  dopasowania wtórnego wierzchołka

527        Dżety różnią się liczbą częstek je tworzących oraz liczbą wtórnego wierzchołków. Większość  
 528        algorytmów uczenia maszynowego wymaga natomiast dostarczenia danych w postaci tabelarycznej (macierzowej), ze stałą liczbą kolumn. Aby spełnić to wymaganie konieczne jest przyjęcie pewnej ustalonej liczby wtórnego wierzchołków oraz częstek tworzących dżet – w przypadku  
 531        gdy dżet ma więcej elementów tego typu są one odrzucane, natomiast puste pola są wypełniane  
 532        zerami w przypadku gdy ma ich mniej. Po przeanalizowaniu rozkładów liczby wtórnego  
 533        wierzchołków i częstek tworzących dżet (Rys. 11) oraz wstępny sprawdzeniu jak dodawanie  
 534        kolejnych elementów wpływa na otrzymywane wyniki (na podstawie wzmacnianych drzew  
 535        decyzyjnych ze względu na wspomnianą w 2.3 szybkość i stabilność) ustalono liczbę częstek  
 536        tworzących dżet równą 15 a wtórnego wierzchołków równą 20.

537        Istotnym zagadnieniem jest także kolejność w jakiej ułożone będą zmienne. Dla sieci kon-  
 538        wolucyjnych szukających lokalnych zależności rozsądne jest ułożenie obok siebie tych samych  
 539        zmiennych, np.  $L_{xy,1}, L_{xy,2}, L_{xy,3} \dots \sigma_{vertex,1}, \sigma_{vertex,2}, \sigma_{vertex,3} \dots$ . Dla sieci w pełni połączonych  
 540        oraz drzew decyzyjnych kolejność zmiennych nie ma znaczenia, ale ważne jest aby ich położenie  
 541        było stałe, np. aby  $L_{xy}$  i  $\sigma_{L_{xy}}$  kolejnego wtórnego wierzchołka były w tych samych miejscach,  
 542        tak aby możliwe było szukanie zależności między nimi.

543        Nastecną kwestią jest wybór wielkości decydującej o kolejności ułożenia elementów, tj. która  
 544        częstka będzie częstką nr 1 a która nr 5. Losowe ułożenie elementów sprawiłoby, że bezpośrednie  
 545        porównywanie wielkości w danych kolumnach (co ma miejsce bezpośrednio w drzewach decy-  
 546        zacyjnych a pośrednio w sieciach neuronowych) straciłoby sens. Z kolei dobór tej kolejności  
 547        pozwala na łatwe odtworzenie przez algorytm uczenia maszynowego motywów fizycznie  
 548        algorytmów omówionych w 1.4.2. Przykładowo cięcie na wartość  $IP$  drugiej lub trzeciej częstki  
 549        (gdy są one posortowane wg malejących wartości  $IP$ ) jest istotą algorytmu nazywanego *Track*

550 Counting – TC stosowanego w CMS i ALICE. Kolejność w jakiej ułożone będą elementy, wpły-  
 551 wa także na to, które z nich będą odrzucone w przypadku gdy dżet zawiera więcej niż 15 cząstek  
 552 i 20 wierzchołków. Ponownie posiliłowano się testami z użyciem drzew decyzyjnych. Ostatecznie  
 553 wtórne wierzchołki ułożono według malejącego  $L_{xy}$  a cząstki – malejącego  $p_T$ .



Rysunek 11: Rozkłady liczby cząstek tworzących dżet i liczby wtórnych wierzchołków (część) wraz wartościami cięć.

## 554 4 Analiza

### 555 4.1 Dobór metryki

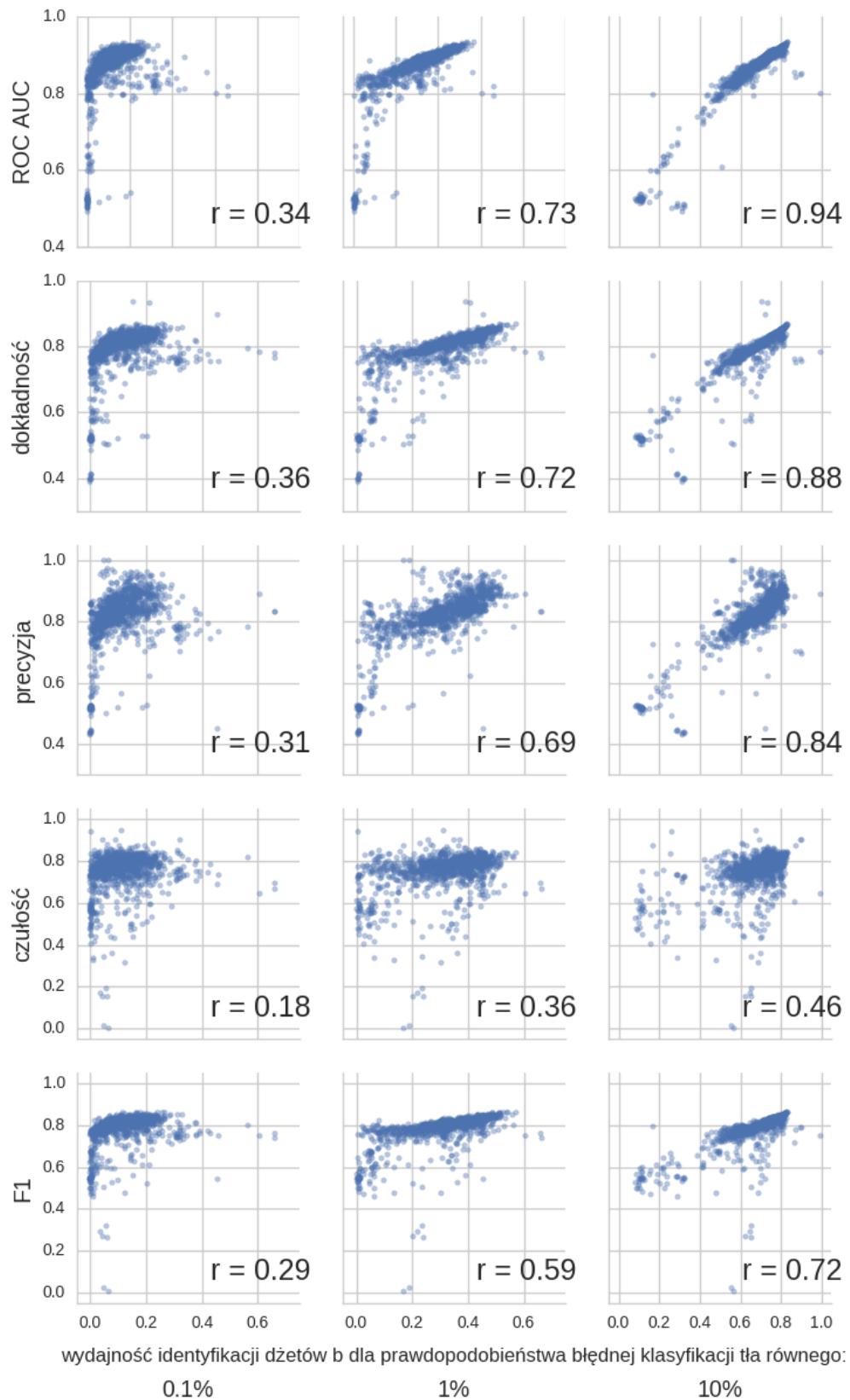
556 Bardzo ważnym elementem w trenowaniu algorytmów uczenia maszynowego jest dobór odpo-  
557 wiedniej metryki – klasycznym złym przykładem jest używanie dokładności (ang. *accuracy*) do  
558 oceniania klasyfikacji binarnej w przypadku dużego niezrównoważenia klas – algorytm przewi-  
559 dujący zawsze klasę większościową może osiągnąć dużą wartość dokładności będąc jednocześnie  
560 bardzo słabym modelem.

561 Kilka najczęściej używanych metryk wymieniono w Tab. ???. Używanie i porównywanie  
562 kilku miar efektywności jest często niepraktyczne dlatego dobrze jest wybrać jedną metrykę.  
563 Przy jej wyborze należy kierować się potencjalnymi zastosowaniami modelu. W tym przypadku  
564 są to analizy fizyczne, które mogą mieć różne wymagania dotyczące czystości i liczebności  
565 otrzymywanych próbek a co za tym idzie, preferować inne punkty pracy zdefiniowane jako pary  
566 liczb: wydajność poprawnej klasyfikacji dżetów  $b$  (ang. *tagging efficiency = true positive rate*  
567 = *recall*), i ułamek niepoprawnie zaklasyfikowanych przypadków tła (ang. *mistagging rate =*  
568 *false positive rate*).

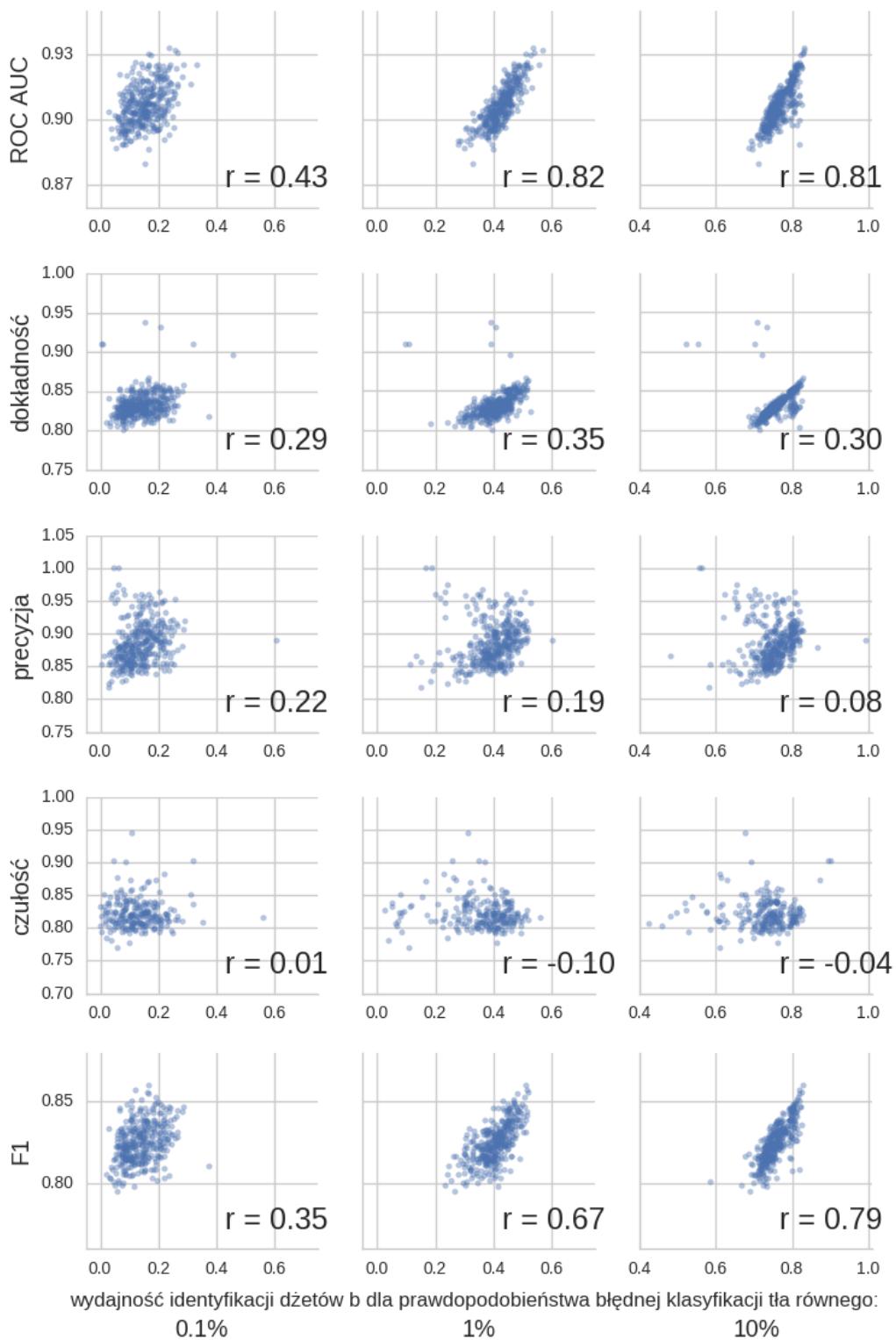
569 Naturalnym wyborem wydaje się pole pod powierzchnią krzywej ROC (ang. *ROC Area*  
570 *Under Curve – ROC AUC*) [50]. Potencjalną przeszkodą może być zakres rozsądnych wartości  
571 prawdopodobieństwa błędnej klasyfikacji przypadków tła: dżety  $b$  stanowią tylko kilka procent  
572 liczby wszystkich dżetów, zatem z punktu widzenia analizy dopuszczalne będą punkty pracy  
573 zapewniające wydajność identyfikacji dżetów  $b$  ok. 10 – 100 razy większą niż częstość niepo-  
574 prawnego zaklasyfikowania przypadków tła. Oznacza to, że zdecydowana większość punktów  
575 pracy znajdujących się na krzywej ROC jest nie do zaakceptowania – interesujące są tylko te o  
576 najniższych częstościach.

577 Aby ilościowo porównywać różne algorytmy wprowadzone zostaną trzy punkty pracy: o  
578 prawdopodobieństwie błędnej klasyfikacji tła równej 0.1%, 1% oraz 10%. Na Rys. 16 przedsta-  
579 wione zostały zależności poszczególnych metryk od wydajności identyfikacji dżetów  $b$  w tych  
580 trzech punktach pracy. Każdy punkt odpowiada jednemu eksperymentowi (dla dowolnego al-  
581 gorytmu) przeprowadzonemu w trakcie przygotowywania analizy. Daje to pogląd, używanie  
582 której metryki zapewni jednocześnie wysokie wartości wydajności na identyfikację dżetów  $b$  w  
583 wybranych punktach pracy. Najwyższe koreacje występują dla punktu pracy o najwyższym  
584 prawdopodobieństwie błędnej klasyfikacji tła – jak będzie to pokazane później wyniki dla tego  
585 punktu pracy są najbardziej stabilne. Spośród analizowanych metryk najwyższe wartości współ-  
586 czynnika Pearsona otrzymano dla pola pod powierzchnią krzywej ROC, dosyć wysokie również  
587 dla dokładności i precyzji. Widać, że wartości czułości są najsłabiej skorelowane z wydajnością  
588 identyfikacji dżetów  $b$  – jest zrozumiałe, że są one dużo niższe niż dla precyzji: wartości czułości  
589 maleją gdy błędnie klasyfikowane są dżety  $b$  stanowiące sygnał, podczas gdy wartości precyzji  
590 maleją, gdy błędnie klasyfikowane są przypadki tła. Z tych dwóch błędów, drugi jest bardziej  
591 kosztowny, gdyż błędna klasyfikacja nawet niewielkiej części tła znacznie pogarsza czystość  
592 otrzymywanej próbki.

593 Na Rys. 13 przedstawiono te same wykresy, ale tym razem wybrano tylko 25% najleps-  
594 szych wartości dla każdej metryki – te punkty są bardziej znaczące, gdyż ostatecznie modele z  
595 eksperymentów dających najlepsze wyniki będą używane. Dla tych wykresów otrzymano zde-  
596 cydowaną dominację *ROC AUC* – wybór modeli dających najwyższe pole pod powierzchnią  
597 krzywej ROC zapewnia jednocześnie otrzymanie wysokich wartości wydajności identyfikacji  
598 dżetów  $b$  dla wybranych punktów pracy.

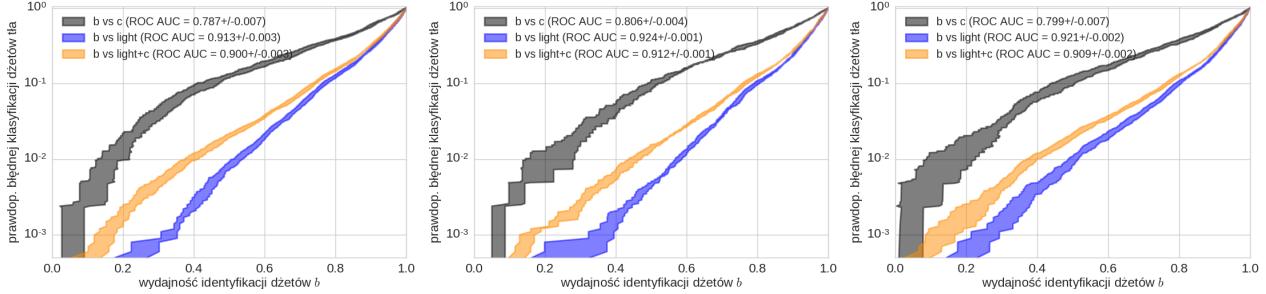


Rysunek 12: Zależność podstawowych metryk od wydajności identyfikacji dżetów  $b$  dla punktów pracy o prawdopodobieństwie błędnej klasyfikacji tła równej 0.1%, 1% oraz 10%. Dla każdego wykresu przedstawiono współczynnik korelacji  $r$  Pearsona.

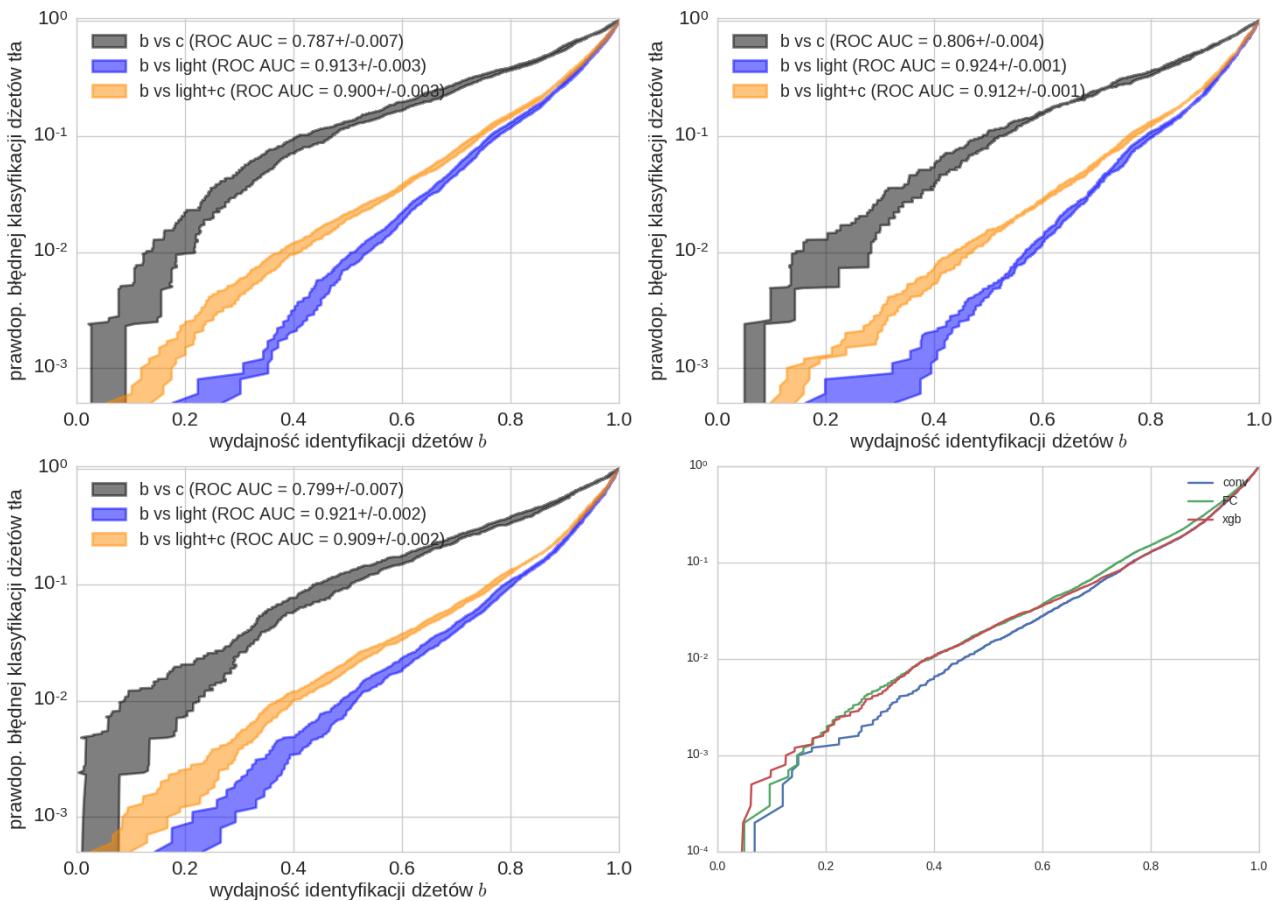


Rysunek 13: Rysunek podobny do Rys. 16, ale przedstawione zostały tylko punkty odpowiadające eksperymentom o wartościach metryki będących w górnym kwartylu wartości danej metryki dla wszystkich eksperymentów.

599 **4.2 Wyniki dla zmiennych SV**

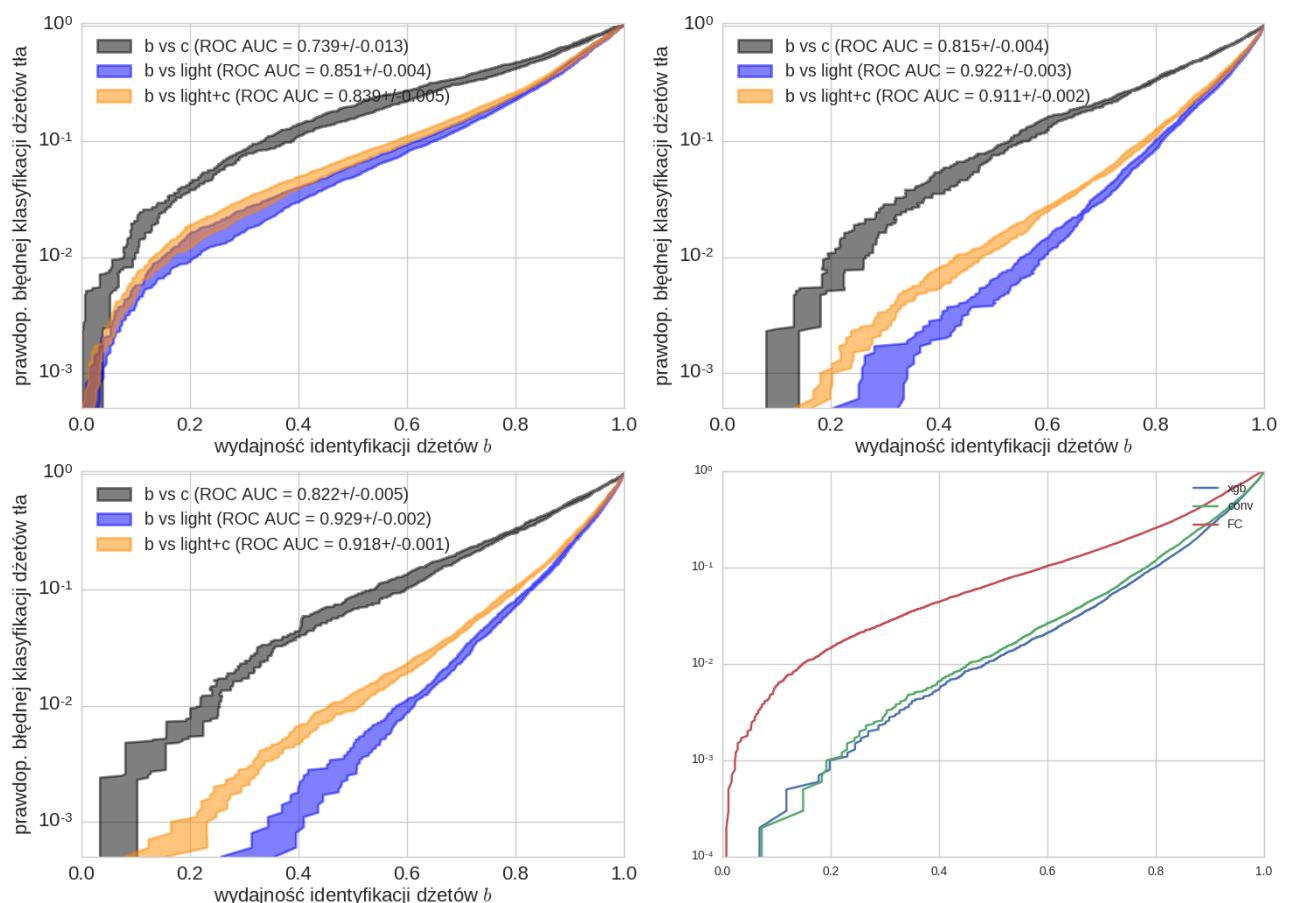


Rysunek 14: Zależności



Rysunek 15: Zależności

600 **4.3 Wyniki dla zmiennych constit**



Rysunek 16: Zależności

## 601 References

- 602 [1] Donald H. Perkins. “Oddziaływanie międzykwasikowe i chromodynamika kwantowa”. In:  
603 *Wstęp do Fizyki Wysokich Energii*. 2nd ed. PWN, 2005, 171–193.
- 604 [2] David J. Gross and Frank Wilczek. “Ultraviolet Behavior of Nonabelian Gauge Theories”.  
605 In: *Phys. Rev. Lett.* 30 (1973). [,271(1973)], pp. 1343–1346. DOI: 10.1103/PhysRevLett.  
606 30.1343.
- 607 [3] H. David Politzer. “Reliable Perturbative Results for Strong Interactions?” In: *Phys. Rev.*  
608 *Lett.* 30 (1973). [,274(1973)], pp. 1346–1349. DOI: 10.1103/PhysRevLett.30.1346.
- 609 [4] C. Patrignani et al. “Review of Particle Physics”. In: *Chin. Phys.* C40.10 (2016), p. 100001.  
610 DOI: 10.1088/1674-1137/40/10/100001.
- 611 [5] John C. Collins and M. J. Perry. “Superdense Matter: Neutrons Or Asymptotically Free  
612 Quarks?” In: *Phys. Rev. Lett.* 34 (1975), p. 1353. DOI: 10.1103/PhysRevLett.34.1353.
- 613 [6] N. Cabibbo and G. Parisi. “Exponential Hadronic Spectrum and Quark Liberation”. In:  
614 *Phys. Lett.* 59B (1975), pp. 67–69. DOI: 10.1016/0370-2693(75)90158-6.
- 615 [7] D. Boyanovsky, H. J. de Vega, and D. J. Schwarz. “Phase transitions in the early and the  
616 present universe”. In: *Ann. Rev. Nucl. Part. Sci.* 56 (2006), pp. 441–500. DOI: 10.1146/  
617 annurev.nucl.56.080805.140539. arXiv: hep-ph/0602002 [hep-ph].
- 618 [8] Mark G. Alford and Kai Schwenzer. “What the Timing of Millisecond Pulsars Can Teach  
619 us about Their Interior”. In: *Phys. Rev. Lett.* 113.25 (2014), p. 251102. DOI: 10.1103/  
620 PhysRevLett.113.251102. arXiv: 1310.3524 [astro-ph.HE].
- 621 [9] Vardan Khachatryan et al. “Evidence for collectivity in pp collisions at the LHC”. In:  
622 *Phys. Lett.* B765 (2017), pp. 193–220. DOI: 10.1016/j.physletb.2016.12.009. arXiv:  
623 1606.06198 [nucl-ex].
- 624 [10] Jaroslav Adam et al. “Enhanced production of multi-strange hadrons in high-multiplicity  
625 proton-proton collisions”. In: *Nature Phys.* 13 (2017), pp. 535–539. DOI: 10.1038/  
626 nphys4111. arXiv: 1606.07424 [nucl-ex].
- 627 [11] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. “The Anti-k(t) jet clustering al-  
628 gorithm”. In: *JHEP* 04 (2008), p. 063. DOI: 10.1088/1126-6708/2008/04/063. arXiv:  
629 0802.1189 [hep-ph].
- 630 [12] Vardan Khachatryan et al. “Charged-particle nuclear modification factors in PbPb and  
631 pPb collisions at  $\sqrt{s_{NN}} = 5.02$  TeV”. In: *JHEP* 04 (2017), p. 039. DOI: 10.1007/  
632 JHEP04(2017)039. arXiv: 1611.01664 [nucl-ex].
- 633 [13] Betty Abelev et al. “Centrality Dependence of Charged Particle Production at Large  
634 Transverse Momentum in Pb–Pb Collisions at  $\sqrt{s_{NN}} = 2.76$  TeV”. In: *Phys. Lett.*  
635 B720 (2013), pp. 52–62. DOI: 10.1016/j.physletb.2013.01.051. arXiv: 1208.2711  
636 [hep-ex].
- 637 [14] Carlos A. Salgado and Urs Achim Wiedemann. “Calculating quenching weights”. In: *Phys.*  
638 *Rev.* D68 (2003), p. 014008. DOI: 10.1103/PhysRevD.68.014008. arXiv: hep-ph/0302184  
639 [hep-ph].
- 640 [15] Yuri L. Dokshitzer and D. E. Kharzeev. “Heavy quark colorimetry of QCD matter”. In:  
641 *Phys. Lett.* B519 (2001), pp. 199–206. DOI: 10.1016/S0370-2693(01)01130-3. arXiv:  
642 hep-ph/0106202 [hep-ph].

- [16] Georges Aad et al. “Performance of  $b$ -Jet Identification in the ATLAS Experiment”. In: *JINST* 11.04 (2016), P04008. DOI: 10.1088/1748-0221/11/04/P04008. arXiv: 1512.01094 [hep-ex].
- [17] A. M. Sirunyan et al. “Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV”. In: *JINST* 13.05 (2018), P05011. DOI: 10.1088/1748-0221/13/05/P05011. arXiv: 1712.07158 [physics.ins-det].
- [18] Serguei Chatrchyan et al. “Identification of b-quark jets with the CMS experiment”. In: *JINST* 8 (2013), P04013. DOI: 10.1088/1748-0221/8/04/P04013. arXiv: 1211.4462 [hep-ex].
- [19] Linus Feldkamp. “Study of b-jet tagging performance in ALICE”. In: *J. Phys. Conf. Ser.* 509 (2014), p. 012061. DOI: 10.1088/1742-6596/509/1/012061. arXiv: 1310.2817 [hep-ex].
- [20] Rüdiger Haake. “Machine and deep learning techniques in heavy-ion collisions with ALICE”. In: *Proceedings, 2017 European Physical Society Conference on High Energy Physics (EPS-HEP 2017): Venice, Italy, July 5-12, 2017*. Vol. EPS-HEP2017. 2017. DOI: 10.22323/1.314.0498. arXiv: 1709.08497 [physics.data-an]. URL: <https://pos.sissa.it/314/498/pdf>.
- [21] Roel Aaij et al. “Identification of beauty and charm quark jets at LHCb”. In: *JINST* 10.06 (2015), P06013. DOI: 10.1088/1748-0221/10/06/P06013. arXiv: 1504.07670 [hep-ex].
- [22] K. Aamodt et al. “The ALICE experiment at the CERN LHC”. In: *JINST* 3 (2008), S08002. DOI: 10.1088/1748-0221/3/08/S08002.
- [23] Betty Bezverkhny Abelev et al. “Performance of the ALICE Experiment at the CERN LHC”. In: *Int. J. Mod. Phys.* A29 (2014), p. 1430044. DOI: 10.1142/S0217751X14300440. arXiv: 1402.4476 [nucl-ex].
- [24] Wikimedia Commons. *Schematics of the ALICE subdetectors*. 2014. URL: [https://commons.wikimedia.org/wiki/File:2012-Aug-02-ALICE\\_3D\\_v0\\_with\\_Text\\_\(1\)\\_2.jpg](https://commons.wikimedia.org/wiki/File:2012-Aug-02-ALICE_3D_v0_with_Text_(1)_2.jpg).
- [25] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. “Supervised machine learning: A review of classification techniques”. In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.
- [26] Marcin Wolter. “Metody analizy wielu zmiennych w fizyce wysokich energii”. PhD thesis. IFJ PAN, 2012.
- [27] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24.2 (1996), pp. 123–140.
- [28] Yoav Freund and Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.
- [29] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: <http://doi.acm.org/10.1145/2939672.2939785>.
- [30] James Bergstra and Yoshua Bengio. “Random Search for Hyper-parameter Optimization”. In: *J. Mach. Learn. Res.* 13 (Feb. 2012), pp. 281–305. ISSN: 1532-4435. URL: <http://dl.acm.org/citation.cfm?id=2188385.2188395>.

- [31] Sandhya Samarasinghe. *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. Auerbach publications, 2016.
- [32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [33] Kurt Hornik. “Approximation capabilities of multilayer feedforward networks”. In: *Neural Networks* 4.2 (1991), pp. 251–257. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(91\)90009-T](https://doi.org/10.1016/0893-6080(91)90009-T). URL: <http://www.sciencedirect.com/science/article/pii/089360809190009T>.
- [34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.
- [35] Petar Veličković. *Deep learning for complete beginners: convolutional neural networks with keras*. 2017. URL: <https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html> (visited on 07/15/2018).
- [36] Andrew Ng. *Convolutional Neural Networks*. 2017. URL: <https://www.coursera.org/learn/convolutional-neural-networks> (visited on 07/15/2018).
- [37] MathWorks. *Convolutional Neural Network*. URL: <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html> (visited on 07/15/2018).
- [38] Kendrick Tan. *Capsule Networks Explained*. 2017. URL: [https://knrdck.co/posts/capsule\\_networks\\_explained/](https://knrdck.co/posts/capsule_networks_explained/) (visited on 07/15/2018).
- [39] François Chollet et al. *Keras*. <https://keras.io>. 2015.
- [40] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [41] Timothy Dozat. *Incorporating Nesterov Momentum into Adam*. 2015. URL: {[http://cs229.stanford.edu/proj2015/054\\_report.pdf](http://cs229.stanford.edu/proj2015/054_report.pdf)}.
- [42] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980. URL: <http://arxiv.org/abs/1412.6980>.
- [43] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *J. Mach. Learn. Res.* 15.1 (Jan. 2014), pp. 1929–1958. ISSN: 1532-4435. URL: <http://dl.acm.org/citation.cfm?id=2627435.2670313>.
- [44] Torbjorn Sjostrand, Stephen Mrenna, and Peter Z. Skands. “A Brief Introduction to PYTHIA 8.1”. In: *Comput. Phys. Commun.* 178 (2008), pp. 852–867. DOI: 10.1016/j.cpc.2008.01.036. arXiv: 0710.3820 [hep-ph].
- [45] Peter Skands, Stefano Carrazza, and Juan Rojo. “Tuning PYTHIA 8.1: the Monash 2013 Tune”. In: *Eur. Phys. J. C*74.8 (2014), p. 3024. DOI: 10.1140/epjc/s10052-014-3024-y. arXiv: 1404.5630 [hep-ph].
- [46] René Brun et al. “GEANT Detector Description and Simulation Tool”. In: (1994). DOI: 10.17181/CERN.MUHF.DMJ1.
- [47] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. “FastJet User Manual”. In: *Eur. Phys. J. C*72 (2012), p. 1896. DOI: 10.1140/epjc/s10052-012-1896-2. arXiv: 1111.6097 [hep-ph].
- [48] D0 Collaboration. *Observation of Single Top Quark Production*. 2009. URL: [https://www-d0.fnal.gov/Run2Physics/top/singletop\\_observation/singletop\\_observation\\_updated.html](https://www-d0.fnal.gov/Run2Physics/top/singletop_observation/singletop_observation_updated.html) (visited on 07/15/2018).

- 733 [49] Matteo Cacciari and Gavin P. Salam. “Pileup subtraction using jet areas”. In: *Phys. Lett.*  
734 B659 (2008), pp. 119–126. doi: 10.1016/j.physletb.2007.09.077. arXiv: 0707.1378  
735 [hep-ph].
- 736 [50] Andrew P Bradley. “The use of the area under the ROC curve in the evaluation of  
737 machine learning algorithms”. In: *Pattern recognition* 30.7 (1997), pp. 1145–1159.