

# 1 Spis treści

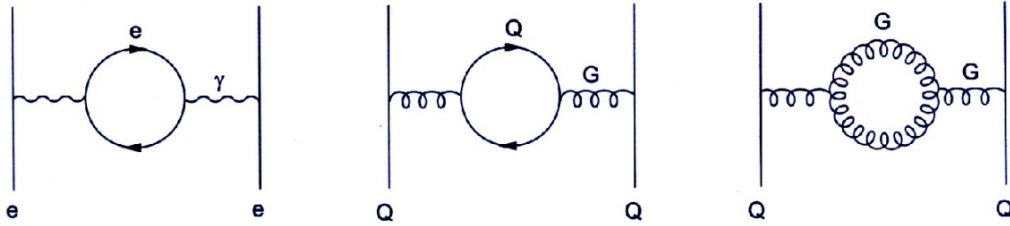
2	<b>1 Fizyka dżetów cząstek</b>	<b>2</b>
3	1.1 Chromodynamika kwantowa . . . . .	2
4	1.2 Plazma kwarkowo-gluonowa . . . . .	3
5	1.3 Dżety . . . . .	3
6	1.4 Dżety $b$ . . . . .	4
7	1.5 Eksperyment ALICE . . . . .	5
8	<b>2 Uczenie maszynowe</b>	<b>9</b>
9	2.1 Wzmacniane drzewa decyzyjne . . . . .	9
10	2.2 Sieci neuronowe . . . . .	10
11	2.3 Dyskusja użycia dwóch algorytmów . . . . .	15
12	<b>3 Dane</b>	<b>16</b>
13	<b>4 Analiza</b>	<b>18</b>
14	4.1 Dobór metryki . . . . .	18
15	4.2 Wyniki dla zmiennych SV . . . . .	21
16	4.3 Wyniki dla zmiennych constit . . . . .	21

# 1 Fizyka dżetów cząstek

## 1.1 Chromodynamika kwantowa

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

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

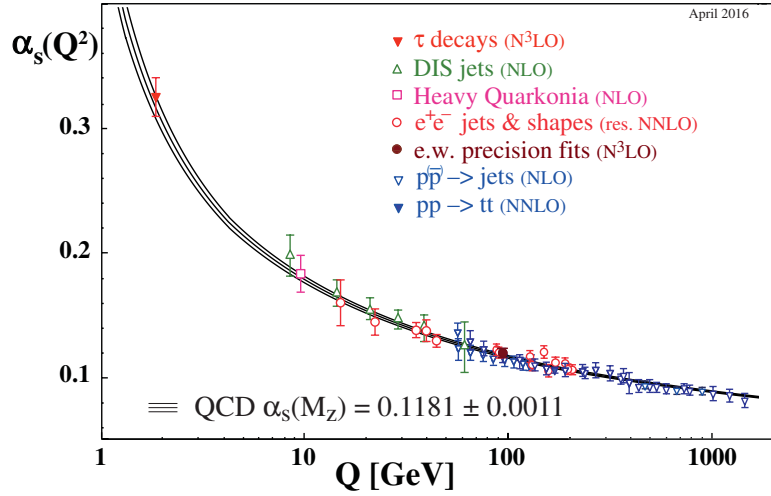


Rysunek 1: Diagramy Feynmana opisujące polaryzację próżni w *QED* (lewy) i *QCD* (środkowy i prawy). Rysunki lewy i środkowy są swoimi odpowiednikami w tych dwóch teoriach, natomiast prawy, w którym oddziałują jedynie bozony pośredniczące nie ma swojego odpowiednika w *QED*. Źródło: [1]

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

- asymptotyczna swoboda (ang. *asymptotic freedom*) [2], [3] – dla wysokich energii ( $\gtrsim 100$  GeV) silna stała sprzężenia jest mała (w tym zakresie energii możliwe jest stosowanie rachunku perturbacyjnego) i kwarki wewnątrz hadronów zachowują się jak cząstki quasi-swobodne.

- uwięzienie koloru (ang. *colour confinement*) – przy zwiększaniu odległości między partonami siła oddziaływania rośnie do nieskończoności, dlatego nigdy nie obserwuje się swobodnych cząstek obdarzonych ładunkiem kolorowym a jedynie związane w kolorowo obojętne hadrony.



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

## 1.2 Plazma kwarkowo-gluonowa

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

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

Cechą charakterystyczną  $QGP$  jest obecność wolnych kwarków i gluonów (zbiórco nazywanych partonami). Ze względu na uwięzienie koloru w każdym innym stanie materii są one zawsze związane i tworzą hadrony. Wolne kwarki i gluony powstające w zderzeniach muszą zatem przejść przez proces hadronizacji, w którym rekombinują one ze spontanicznie wytwarzanymi nowymi partonami, tworząc hadrony. W wyniku tego procesu, z każdego partonu obecnego w początkowym etapie zderzenia może powstać wiele cząstek poruszających się podobnym kierunkiem, tworząc stożek z wierzchołkiem blisko punktu interakcji wiązek. Taki stożek skolimowanych cząstek nazywany jest dżetem cząstek.

## 1.3 Dżety

Przedstawiona powyżej definicja dżetu nie jest precyzyjna z punktu widzenia pracy eksperymentalnej. W detektorze obserwuje się tylko cząstki w stanie końcowym, nie jest zatem możliwe

81 przyporządkowanie cząstki według jej pochodzenia. W związku z tym, konieczne jest użycie al-  
82 gorytmu klasteryzującego, dostającego na wejściu tylko obserwowalne eksperymentalnie cząst-  
83 ki. To jakie dzęty 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 Eksperymentalne 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 zostałaby cecha odpowiedniości między obiektami 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 eksperymentalnych.

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 przecho-  
97 dzący 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 pochodzącymi 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 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.

## 113 1.4 Dżety $b$

### 114 Właściwości

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

120 Identyfikacja dżetów  $b$  jest sporym wyzwaniem ze względu na zdecydowanie częściej wy-  
121 stępujące dżety lekkie, tj. powstałe z hadronizacji kwarków  $u, d, s$  lub gluonów. Rozpoznawanie  
122 dżetów  $b$  bazuje na charakterystycznych właściwościach hadronów zawierających kwark piękny:  
123 relatywnie długim czasie życia oraz (w mniejszym stopniu) na ich pół-leptonowych rozpadach  
124 o względnej częstości rozpadu w tym kanale (ang. *branching ratio*) na poziomie 10%.

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

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

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

Algorytmy wykorzystujące informację o poszczególnych cząstkach mogą zasadniczo bazować albo na sumie logarytmów prawdopodobieństw pochodzenia każdej cząstki z pierwotnego wierzchołka (por. ATLAS: IP3D, CMS: JP) lub tym samym prawdopodobieństwie ale dla wybranej, np. drugiej lub trzeciej cząstki na liście posortowanej według malejącego  $IP$  (por. ALICE i CMS: TC). Bardziej złożonym podejściem, w którym cząstki nie są traktowane jako niezależne, jest użycie rekurencyjnych sieci neuronowych (por. ATLAS: RNNIP).

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

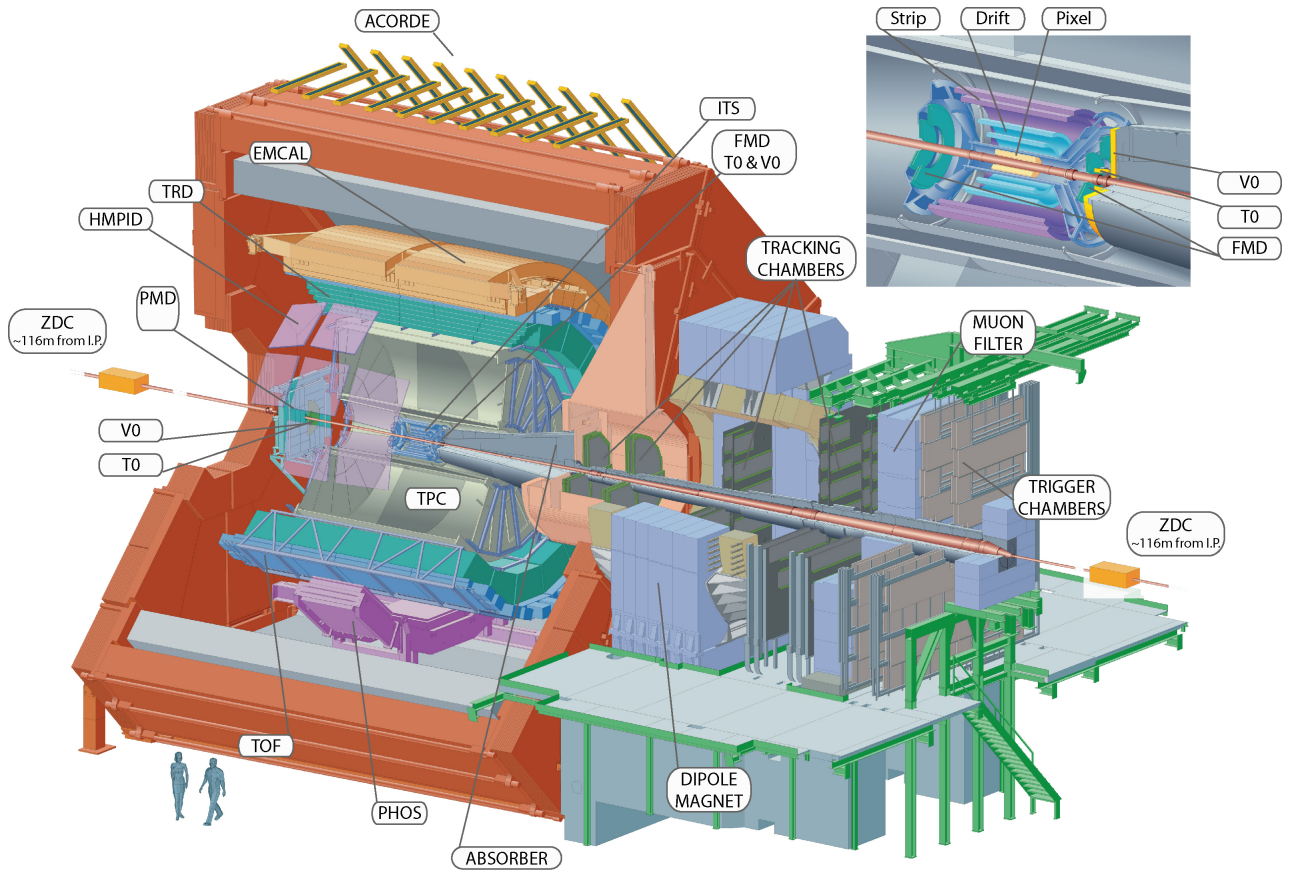
Znaczącą poprawę zdolności predykcyjnej można uzyskać łącząc kilka różnych modeli. Algorytm łączący może pobierać na wejściu albo tylko predykcje klasyfikatorów niższego poziomu (CMS: cMVA<sub>v2</sub>) lub dodatkowo także ich zmienne wejściowe (ATLAS: MV2, DL1). Do scalania używane są zwykle wzmacniane drzewa decyzyjne lub sieci neuronowe.

Przykład innego podejścia zaprezentowała współpraca przy eksperymencie LHC<sub>b</sub>, gdzie wykorzystano dwa zestawy wzmacnianych drzew decyzyjnych operujących na zmiennych związanych z wtórnymi wierzchołkami. Pierwszy zapewnia separację dżetów lekkich od ciężkich a drugi odróżnia dżety  $b$  od  $c$ . Do wyboru punktu pracy, zamiast jednowymiarowego rozkładu predykcji używany jest dwuwymiarowy rozkład wag przypisany przez oba klasyfikatory [21].

## 1.5 Eksperyment ALICE

Eksperyment ALICE [22], [23] jest jednym z czterech największych eksperymentów na LHC. Jest on dedykowany zderzeniom ciężkich jonów (w LHC są to jony ołowiu PbPb), ale mierzone są także mniejsze systemy, tj. proton-proton pp (głównie jako referencję dla pomiarów PbPb) oraz proton-ołów p+Pb, które dostarczają także okazji do badania asymetrycznych zderzeń. Cechą charakterystyczną pomiarów ciężkojonowych jest ich znacznie większa niż w przypadku zderzeń pp krotność, tj. liczba cząstek wyprodukowana w pojedynczym zderzeniu. W przypadku zderzeń PbPb może powstawać nawet do 8000 naładowanych cząstek na jednostkę pseudorapidity  $\eta$  (ang. *pseudorapidity*)<sup>1</sup>. Detektor ALICE został zoptymalizowany do mierzeniach takich przypadków, jak również pod kątem rekonstrukcji i identyfikacji cząstek o szerokim zakresie pędów (100 MeV – 100 GeV).

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



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

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

- Detektory śladowe – mierzące trajektorie cząstek zakrzywiane w polu magnetycznym o wartości  $B = 0.5$  T.

- Inner Tracking System (ITS) – zespół krzemowych detektorów śladowych znajdujący się najbliżej miejsca interakcji wiązek. Składa się on z 6 cylindrycznych warstw o promieniach od 4 do 43 cm, wykonanych w trzech różnych technologiach. Jego główną rolą jest rekonstrukcja pierwotnego oraz wtórnych wierzchołków. Bierze także udział w rekonstrukcji trajektorii i strat energetycznych cząstek, szczególnie tych niskopędowych, które nie docierają do dalej położonych detektorów.

- Time Projection Chamber (TPC) – długa na 5m i o takiej średnicy komora projekcji czasowej. Jest to główny detektor śladowy ALICE, wraz z ITS służy do wyznaczania trajektorii cząstek i na ich podstawie również wierzchołków zderzenia. Elektrony uwolnione ze zjonizowanego przez poruszające się w nim naładowane cząstki gazu dryfują wzdłuż kierunku wiązki w stronę końcowych elektrod. Następnie są tam zbierane dostarczając informacji o dwóch współrzędnych toru cząstki: odległości od wiązki i kącie azymutalnym. Trzecia składowa trajektorii jest otrzymywana na podstawie czasu dotarcia elektronów do elektrod. TPC jest najwolniejszym detektorem ALICE (ze względu na ograniczający czas dryfu elektronów wynoszący  $\sim 90 \mu s$ ), użycie detektora tego typu podyktowane jest jego zdolnością do rozwikłania śladów tysięcy cząstek spodziewanych w centralnych zderzeniach PbPb.

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

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

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

- kalorymetry

- Photon Spectrometer (PHOS) – elektromagnetyczny kalorymetr o wysokiej rozdzielczości energetycznej i przestrzennej (podzielony na kryształy o rozmiarze poprzecznym  $2.2 \times 2.2$  cm, co odpowiada rozmiarowi w dziedzinie  $\eta$ ,  $\phi$   $0.004 \times 0.004$ ). Pokrywa zakres pseudospieszości  $|\eta| < 0.12$  i kąta azymutalnego równy  $100^\circ$ . PHOS ma za zadanie identyfikację i pomiar czteropędów fotonów, w szczególności tych nie pochodzących z rozpadu innych cząstek (ang. *direct photons*) oraz lekkich mezonów neutralnych (np.  $\pi^0$ ) przez dwufotonowy kanał rozpadu.
- Electromagnetic Calorimeter (EMCal) – drugi elektromagnetyczny kalorymetr ALICE o mniejszej ziarnistości ( $\Delta\eta, \Delta\phi = 0.014 \times 0.014$ ), ale dużo większej akceptancji ( $|\eta| < 0.7$ ,  $\Delta\phi = 107^\circ$ ). EMCal poprawia możliwości ALICE w zakresie pomiarów tłumienia dżetów, pozwalając na wyznaczanie neutralnej składowej energii dżetów (energii niesionej przez neutralne cząstki). Dzięki innej charakterystyce dla elektronów i hadronów (elektrony typowo deponują niemal całą energię a hadrony tylko niewielką część) pozwala je odróżnić na podstawie stosunku zmierzonej w nim energii do wyznaczonego wcześniej pędu  $E/p$ . EMCal może być użyty także w szybkim systemie wyzwalania, do selekcji przypadków z dżetami oraz wysokoenergetycznymi fotonami i elektronami.

- Muon spectrometer – spektrometr mionowy, złożony z dwóch pasywnych absorberów, znajdujących się między nimi 10 warstw detektora śladowego oraz komór systemu wyzwalającego 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 powstałymi 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 precyzyjnie 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 resztkami gazu obecnymi w rurach próżniowych.
- T0 – dostarcza dokładny czas interakcji potrzebny dla detektora TOF, pozwala także na śledzenie świetlności w czasie rzeczywistym.



## 2 Uczenie maszynowe

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

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. *pattern recognition*).

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

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

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

### 2.1 Wzmacniane drzewa decyzyjne

Wzmacniane drzewa decyzyjne są jednym z rozwinięć klasycznego algorytmu drzewa decyzyjnego. Pojedyncze drzewo decyzyjne dzieli przestrzeń cech uczących przy pomocy prostopadłych cięć, na mniejsze/większe niż zadaną wartość w przypadku zmiennej ciągłej lub na należące/nie 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 były jak najbardziej jednorodne. Stosuje się różne miary nieporządku takie jak: indeks Gini  $G_L$  lub entropia  $S_L$ <sup>3</sup>.

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

*Bagging* – w zastosowaniu dla drzew decyzyjnych nazywany algorytmem lasów losowych (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  $L$

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

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

## 309 Parametry i sposób trenowania drzew decyzyjnych na analizowanych danych

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

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

326 Parametry optymalizowane w opisanym procesie to:

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

## 337 2.2 Sieci neuronowe

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

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

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

## 349 Sieci w pełni połączone

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

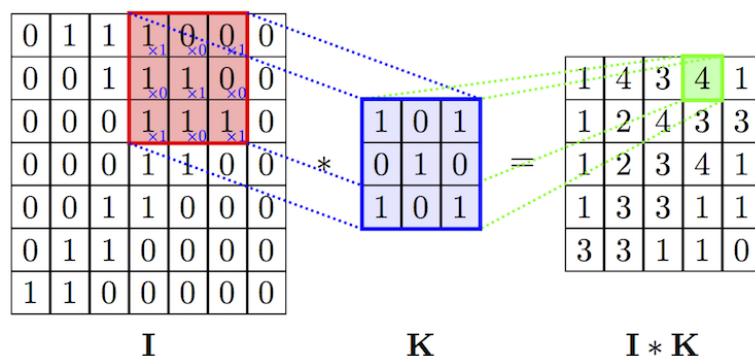
359 Istnieje twierdzenie o sieciach neuronowych jako uniwersalnych aproksymatorach funkcji  
360 (ang. *universal approximation theorem*) [33], mówiące, że już sieć neuronowa o jednej warstwie  
361 ukrytej jest zdolna do przybliżenia dowolnej funkcji z dowolną dokładnością. Twierdzenie to  
362 nie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

## 367 Sieci konwolucyjne

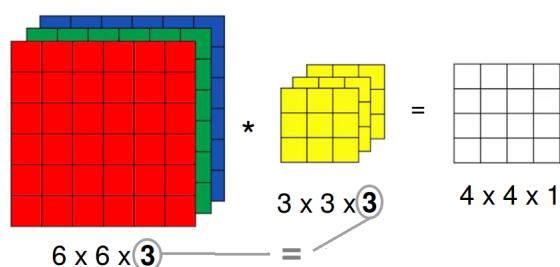
368 Jednym ze sposobów na ograniczenie liczby trenowanych parametrów jest użycie konwolucyj-  
369 nych sieci neuronowych [34] (bardziej poprawną choć rzadko używaną nazwą w języku polskim  
370 jest sieć splotowa). Są one inspirowane połączeniami w korze wzrokowej zwierząt i wywodzą  
371 się z badań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

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

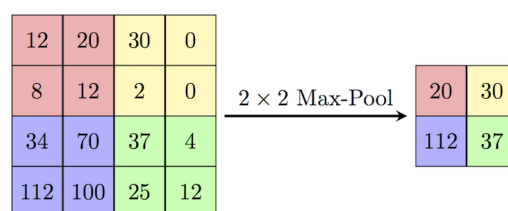
381 W przypadku gdy zamiast wejścia dwuwymiarowego (jak np. obraz czarno-biały), mamy do  
382 czynienia z wejściem trójwymiarowym (np. trzeci wymiar to kolejne kolory w kodowaniu RGB),  
383 filtry również muszą mieć trzy wymiary, przy czym rozmiar w ostatnim wymiarze musi być  
384 równy rozmiarowi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



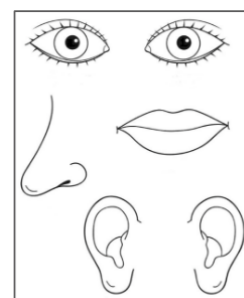
Rysunek 6: Działanie warstwy typu *max-pool*. Źródło: [35].

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.

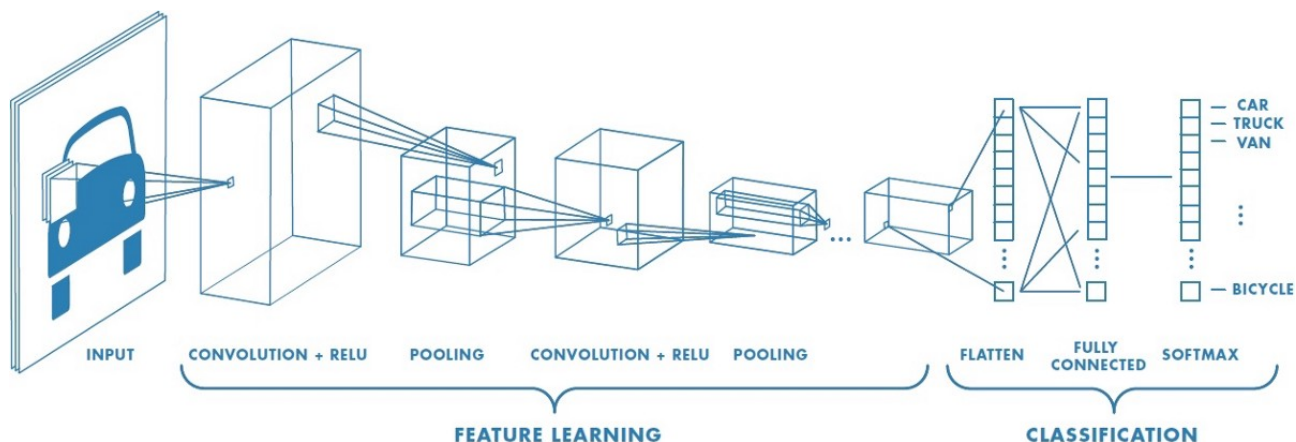
Typowa architektura stosowana w przypadku sieci konwolucyjnych jest następująca: najpierw warstwy konwolucyjne (pomiędzy nimi czasem warstwy typu *max-pool*), następnie wszystkie filtry są rozwijane i składane w długi jednowymiarowy wektor, który przekazywany jest do warstw typu *FC*. W przypadku problemu 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].

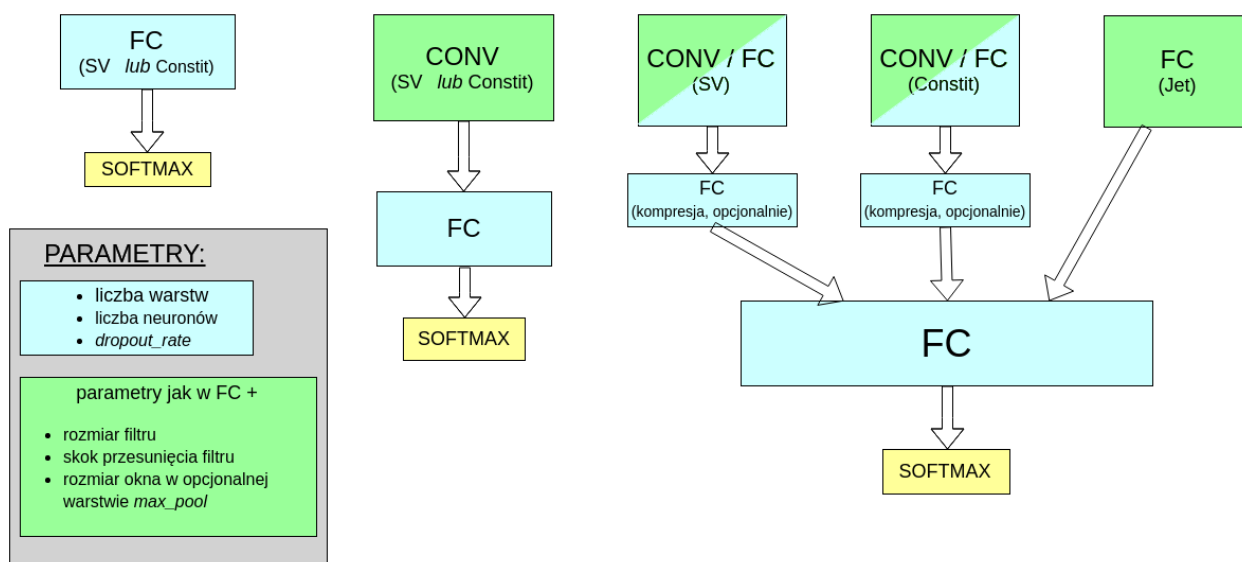


Rysunek 7: Typowa struktura stosowana w sieciach konwolucyjnych. Warstwy konwolucyjne mają za zadanie wydobywać cechy, na podstawie których późniejsze warstwy dokonują klasyfikacji. Widoczna jest charakterystyczna stopniowa zmiana rozmiaru przekazywanej macierzy: rozmiar poprzeczny maleje kosztem głębokości,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

## Hiperparametry i trenowanie sieci neuronowych na analizowanych danych

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

Testowane były trzy architektury: sieci w pełni połączone, sieci konwolucyjne oraz sieć złożona z dwóch gałęzi, osobnych dla wtórnych wierzchołków i cząstek tworzących dżet (por. Rozdz. 3) przedstawione schematycznie na Rys. 9. Do trenowania sieci wykorzystano wysokopoziomową bibliotekę Keras [39] korzystającą z silnika obliczeniowego zaimplementowanego w 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.

Zestaw hiperparametrów definiujący działanie sieci w pełni połączonej:

- 418 • liczba warstw ukrytych
- 419 • liczba neuronów w każdej warstwie
- 420 • funkcja aktywacji – nieliniowa funkcja aplikowana przed zwróceniem wartości w każdym
- 421 neuronie, najpopularniejsze to *tanh*, *ReLU* ( $f(x) = \max(0, x)$ ) oraz funkcja sigmoidalna
- 422 ( $f(x) = \frac{1}{1+\exp(x)}$ )
- 423 • algorytm optymalizacyjny – spadek gradientowy lub jego wariacje
- 424 • parametr szybkości uczenia i jego modyfikacje w trakcie uczenia
- 425 • liczba przykładów trenujących przetwarzanych w jednym kroku uczenia (ang. *batch\_size*)
- 426 – im większy tym szybsze jest trenowanie sieci (dzięki wydajnym operacjom macierzowym), natomiast może się to odbywać kosztem precyzji
- 427
- 428 • liczba epok uczenia – ile razy będzie pokazywany sieci każdy przykład
- 429 • opcjonalnie: warunki stopu (ang. *early stopping*) w razie osiągnięcia *plateau* (wysycenia
- 430 procesu uczenia)
- 431 • opcjonalnie: regularyzacja przy pomocy różnych technik (zwykle konieczna)
- 432 • ponadto dla sieci konwolucyjnych:
  - 433 – liczba warstw konwolucyjnych i liczba filtrów w każdej warstwie
  - 434 – obecność lub brak warstw *max-pool* i rozmiar ich okna
  - 435 – rozmiar filtrów i długość skoku przy ich przesuwaniu
- 436 Same dwie pierwsze wielkości dają nieograniczoną liczbę konfiguracji. Czas trenowania sieci
- 437 neuronowych jest rzędu wielkości większy niż drzew decyzyjnych, dlatego przyjęto szereg kro-
- 438 ków mających na celu zmniejszenie przeszukiwanej przestrzeni hiperparametrów. Na podstawie
- 439 wstępnych testów oraz różnych wskazówek dostępnych w literaturze przyjęto:
  - 440 • *batch\_size* zawsze równy 64 (inne testowane wartości: 16, 32, 128)
  - 441 • za algorytm optymalizacyjny przyjęto algorytm o nazwie *Nadam* [41], tj. rozwinięcie al-
  - 442 gorytmu *Adam* [42] o parametr Nesterova (inne testowane to zwykły spadek gradientowy
  - 443 oraz *Adam*)
  - 444 • funkcję aktywacji: *ReLU*
  - 445 • liczba epok równa 50, 100 lub 200 (lub mniej do testów), zrezygnowano z *early stopping*
  - 446 • stałe w trakcie treningu wartości parametru szybkości uczenia
  - 447 • spośród technik regularyzacyjnych testowano wyłącznie *dropout* [43] z prawdopodobień-
  - 448 stwem odrzucenia równym 0.1, 0.2 lub 0.5
  - 449 • kilka wybranych kombinacji dla zestawu parametrów: rozmiar filtra, długość skoku i roz-
  - 450 miar okna w warstwach *max-pool* – takie same w kolejnych warstwach
  - 451 • liczby neuronów/filtrów w warstwach będące zawsze potęgami dwójki oraz stałą liczbę w
  - 452 kolejnych warstwach lub zmieniającą się o stały czynnik, np. 256-128-64, 128-128-128 lub
  - 453 16-32-64

454 • liczba warstw *FC*: 2-8, konwolucyjnych 2-6

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

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

## 464 2.3 Dyskusja użycia dwóch algorytmów

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

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

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



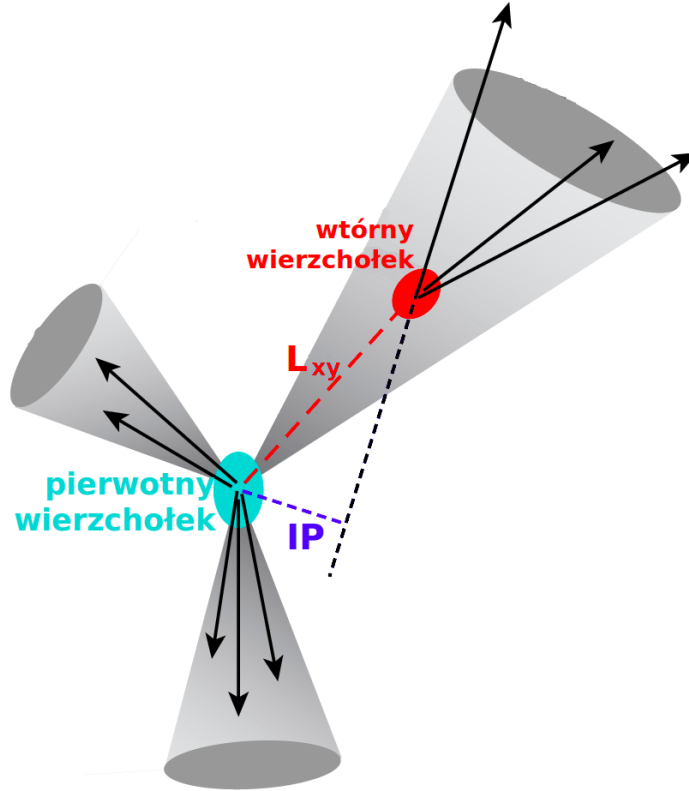
### 3 Dane

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

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

Do analizy wybrano dżety o  $p_T$  większym niż 15 GeV i mieszczących się w całości w akceptancji detektora TPC, tj.  $|\eta| < 0.9$ , co przy użytym parametrze rozmiaru dżetu  $R = 0.4$ , daje ograniczenie na pseudospieszość  $|\eta| < 0.5$  dla osi dżetu.

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



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

Lista używanych zmiennych:

- Zmienne na poziomie dżetu:

- $\eta, \phi$  – pseudospieszość i kąt azymutalny

–  $p_T$  – pęd poprzeczny dżetu

– masa dżetu

– powierzchnia dżetu – liczona w płaszczyźnie  $(\eta, \phi)$ , do powierzchni dżetu zaliczany jest element w powierzchni w którym dodanie cząstki o nieskończenie małym pędzie poprzecznym sprawi, że zostanie ona zaliczona do tego dżetu [#REF <https://arxiv.org/pdf/0707.1378.pdf>]

– gęstość tła (w danym zdarzeniu)

–  $N_{SV}$  – liczba wtórnych wierzchołków

–  $N_{Constit}$  – liczba cząstek tworzących dżet

• Zmienne opisujące cząstki tworzące dżet:

–  $\eta, \phi$  – pseudopospieszość i kąt azymutalny cząstki względem osi dżetu

–  $p_T$  – pęd poprzeczny cząstki

–  $IP_D$  – rzut na kierunek poprzeczny wektora parametru zderzenia

–  $IP_Z$  – rzut na oś  $z$  wektora parametru zderzenia

• Zmienne opisujące wtórne wierzchołki:

–  $L_{xy}$  – odległość między pierwotnym a wtórnym wierzchołkiem (ang. *decay length*)

–  $\sigma_{Lxy}$  – niepewność wyznaczenia  $L_{xy}$

–  $\sigma_{vertex} = \sqrt{d_1^2 + d_2^2 + d_3^2}$  – rozrzut śladów (ang. *tracks*) wokół wtórnego wierzchołka, gdzie  $d_i$  to odległość najbliższego zbliżenia śladu / odległość najbliższego przelotu do wtórnego wierzchołka (ang. *distance of closest approach – DCA*)

–  $M_{inv} = \sqrt{(E_1 + E_2 + E_3)^2 - (\vec{p}_1 + \vec{p}_2 + \vec{p}_3)^2}$  – masa niezmiennicza wierzchołka, gdzie  $E_i, p_i$  to energia i pęd  $i$ -tej cząstki tworzącej wierzchołek

–  $\chi^2/Ndf$  dopasowania wtórnego wierzchołka

TBD: tabelaryczna postać, liczba SV i Nconstit, jak posortowane, wypieniania zerami korespondencja między użytymi danymi a algorytmami z sec. "identyfikacja dżetów b"

## 4 Analiza

### 4.1 Dobór metryki

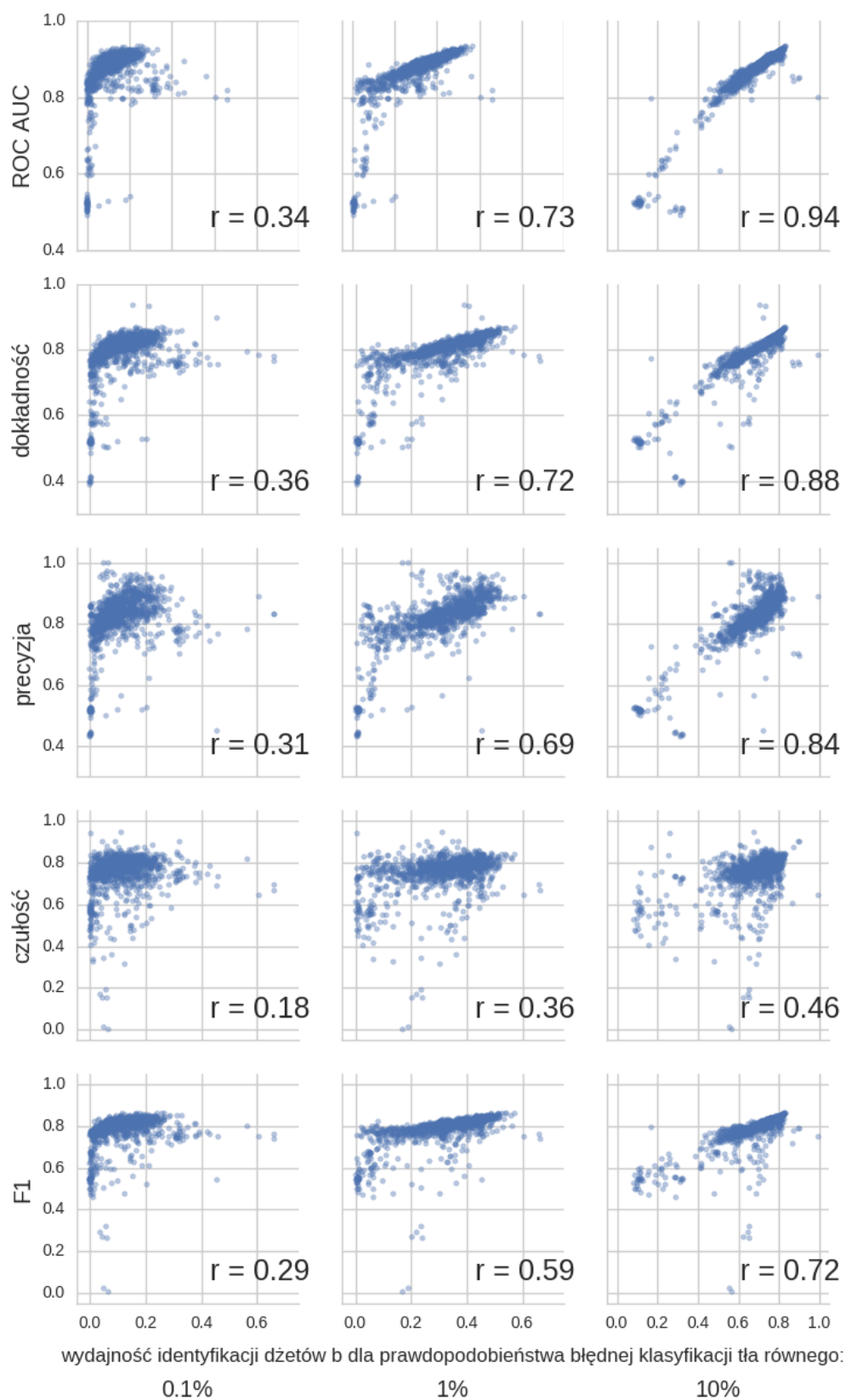
Bardzo ważnym elementem w trenowaniu algorytmów uczenia maszynowego jest dobór odpowiedniej metryki – klasycznym złym przykładem jest używanie dokładności (ang. *accuracy*) do oceniania klasyfikacji binarnej w przypadku dużego niezrównoważenia klas – algorytm przewidyjący zawsze klasę większościową może osiągnąć dużą wartość dokładności będąc jednocześnie bardzo słabym modelem.

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

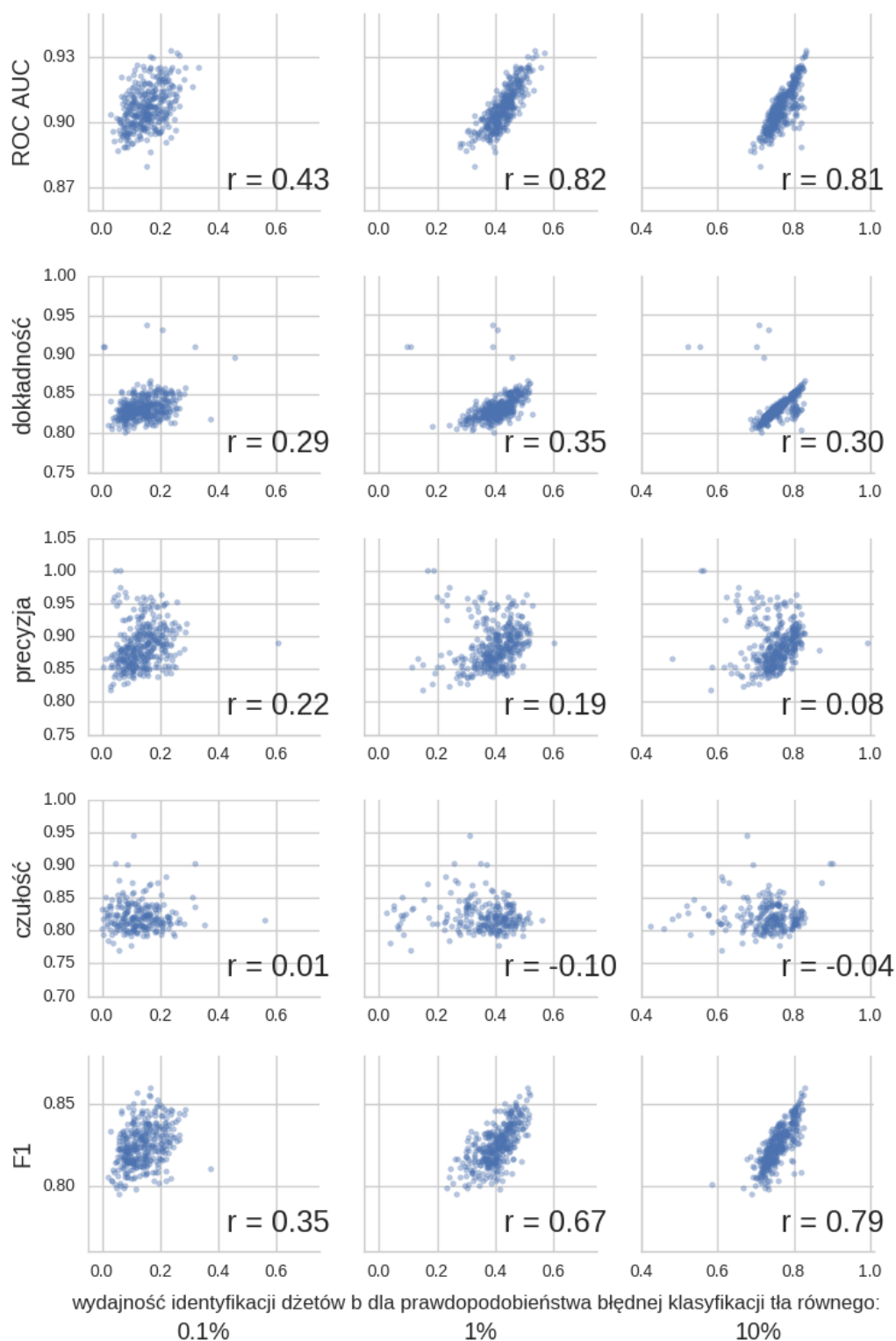
Naturalnym wyborem wydaje się pole pod powierzchnią krzywej ROC (ang. *ROC Area Under Curve – ROC AUC*) [49]. Potencjalną przeszkodą może być zakres rozsądnych wartości prawdopodobieństwa błędnej klasyfikacji przypadków tła: dżety  $b$  stanowią tylko kilka procent liczby wszystkich dżetów, zatem z punktu widzenia analizy dopuszczalne będą punkty pracy zapewniające wydajność identyfikacji dżetów  $b$  ok. 10 – 100 razy większą niż częstość niepoprawnego zaklasyfikowania przypadków tła. Oznacza to, że zdecydowana większość punktów pracy znajdujących się na krzywej ROC jest nie do zaakceptowania – interesujące są tylko te o najniższych częstościach.

Aby ilościowo porównywać różne algorytmy wprowadzone zostaną trzy punkty pracy: o prawdopodobieństwie błędnej klasyfikacji tła równej 0.1%, 1% oraz 10%. Na Rys. 15 przedstawione zostały zależności poszczególnych metryk od wydajności identyfikacji dżetów  $b$  w tych trzech punktach pracy. Każdy punkt odpowiada jednemu eksperymentowi (dla dowolnego algorytmu) przeprowadzonemu w trakcie przygotowywania analizy. Daje to pogląd, używanie której metryki zapewni jednocześnie wysokie wartości wydajności na identyfikację dżetów  $b$  w wybranych punktach pracy. Najwyższe korelacje występują dla punktu pracy o najwyższym prawdopodobieństwie błędnej klasyfikacji tła – jak będzie to pokazane później wyniki dla tego punktu pracy są najbardziej stabilne. Spośród analizowanych metryk najwyższe wartości współczynnika Pearsona otrzymano dla pola pod powierzchnią krzywej ROC, dosyć wysokie również dla dokładności i precyzji. Widać, że wartości czułości są najsłabiej skorelowane z wydajnością identyfikacji dżetów  $b$  – jest zrozumiałe, że są one dużo niższe niż dla precyzji: wartości czułości maleją gdy błędnie klasyfikowane są dżety  $b$  stanowiące sygnał, podczas gdy wartości precyzji maleją, gdy błędnie klasyfikowane są przypadki tła. Z tych dwóch błędów, drugi jest bardziej kosztowny, gdyż błędna klasyfikacja nawet niewielkiej części tła znacznie pogarsza czystość otrzymywanej próbki.

Na Rys. 12 przedstawiono te same wykresy, ale tym razem wybrano tylko 25% najlepszych wartości dla każdej metryki – te punkty są bardziej znaczące, gdyż ostatecznie modele z eksperymentów dających najlepsze wyniki będą używane. Dla tych wykresów otrzymano zdecydowaną dominację *ROC AUC* – wybór modeli dających najwyższe pole pod powierzchnią krzywej ROC zapewnia jednocześnie otrzymanie wysokich wartości wydajności identyfikacji dżetów  $b$  dla wybranych punktów pracy.

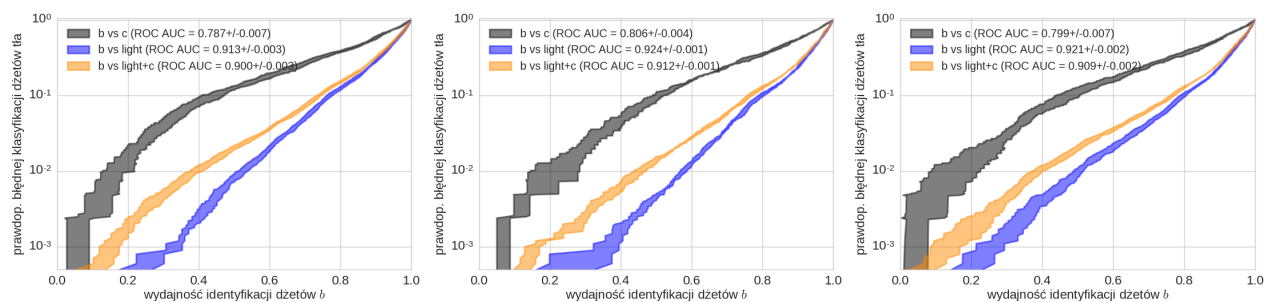


Rysunek 11: Zależność podstawowych metryk od wydajności identyfikacji dżetów  $b$  dla punktów pracy o prawdopodobieństwie błędnej klasyfikacji tła równej 0.1%, 1% oraz 10%. Dla każdego wykresu przedstawiono współczynnik korelacji  $r$  Pearsona.

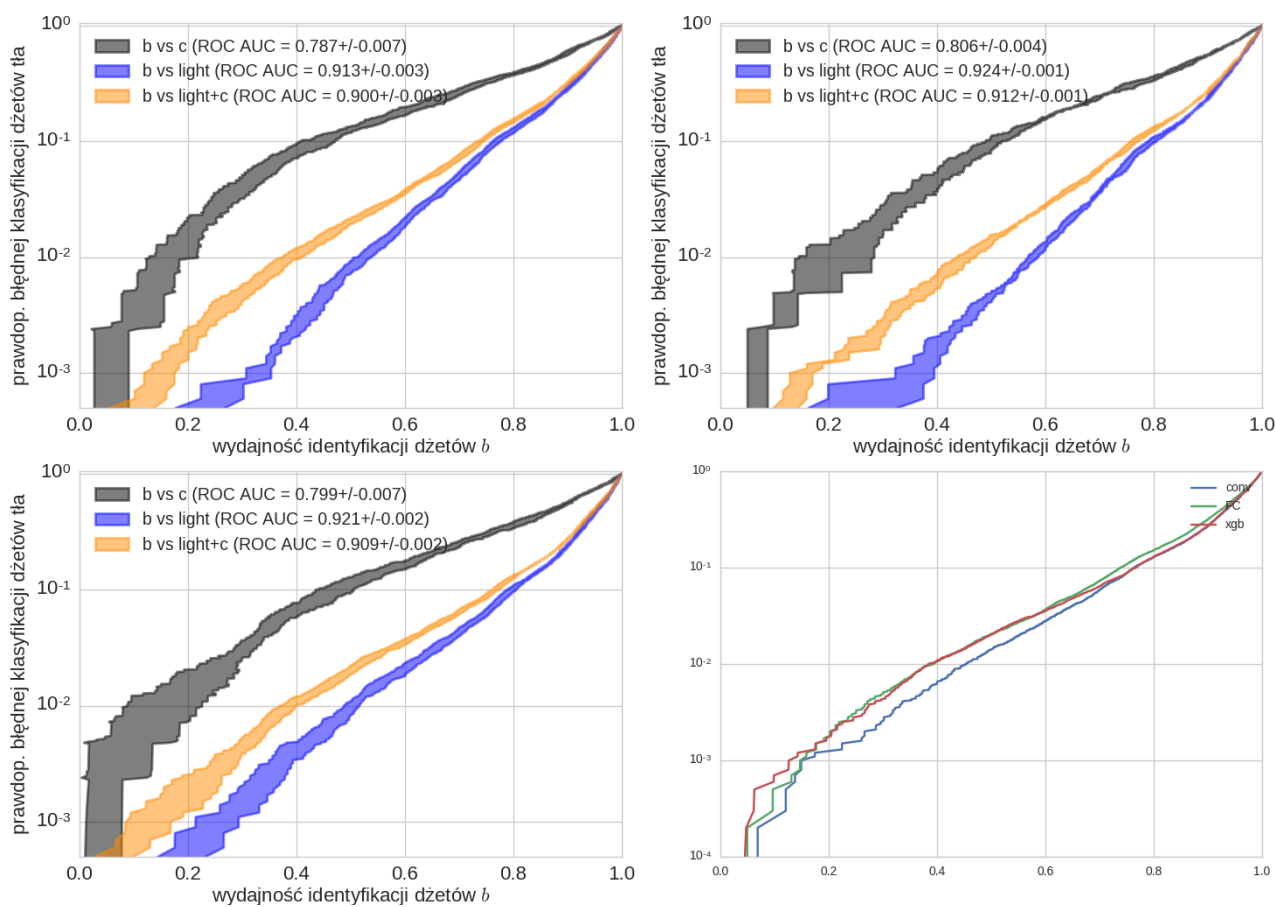


Rysunek 12: Rysunek podobny do Rys. 15, ale przedstawione zostały tylko punkty odpowiadające eksperymentom o wartościach metryki będących w górnym kwartylu wartości danej metryki dla wszystkich eksperymentów.

## 4.2 Wyniki dla zmiennych SV

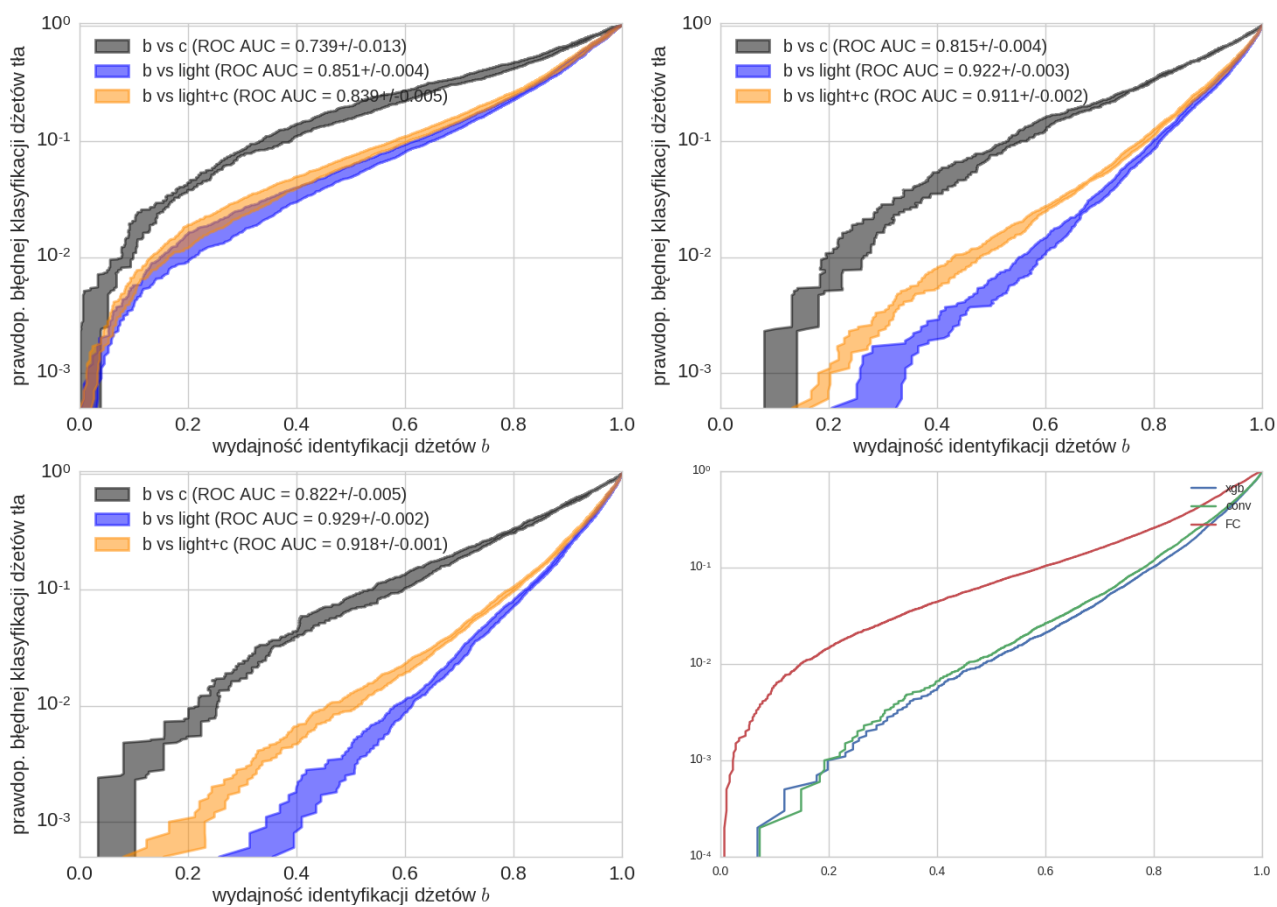


Rysunek 13: Zależności



Rysunek 14: Zależności

## 4.3 Wyniki dla zmiennych constit



Rysunek 15: Zależności



## References

- 576 [1] Donald H. Perkins. “Oddziaływania międzykwarkowe i chromodynamika kwantowa”. In:  
577 *Wstęp do Fizyki Wysokich Energii*. 2nd ed. PWN, 2005, 171–193.
- 578 [2] David J. Gross and Frank Wilczek. “Ultraviolet Behavior of Nonabelian Gauge Theories”.  
579 In: *Phys. Rev. Lett.* 30 (1973). [,271(1973)], pp. 1343–1346. DOI: 10.1103/PhysRevLett.  
580 30.1343.
- 581 [3] H. David Politzer. “Reliable Perturbative Results for Strong Interactions?” In: *Phys. Rev.*  
582 *Lett.* 30 (1973). [,274(1973)], pp. 1346–1349. DOI: 10.1103/PhysRevLett.30.1346.
- 583 [4] C. Patrignani et al. “Review of Particle Physics”. In: *Chin. Phys.* C40.10 (2016), p. 100001.  
584 DOI: 10.1088/1674-1137/40/10/100001.
- 585 [5] John C. Collins and M. J. Perry. “Superdense Matter: Neutrons Or Asymptotically Free  
586 Quarks?” In: *Phys. Rev. Lett.* 34 (1975), p. 1353. DOI: 10.1103/PhysRevLett.34.1353.
- 587 [6] N. Cabibbo and G. Parisi. “Exponential Hadronic Spectrum and Quark Liberation”. In:  
588 *Phys. Lett.* 59B (1975), pp. 67–69. DOI: 10.1016/0370-2693(75)90158-6.
- 589 [7] D. Boyanovsky, H. J. de Vega, and D. J. Schwarz. “Phase transitions in the early and the  
590 present universe”. In: *Ann. Rev. Nucl. Part. Sci.* 56 (2006), pp. 441–500. DOI: 10.1146/  
591 annurev.nucl.56.080805.140539. arXiv: hep-ph/0602002 [hep-ph].
- 592 [8] Mark G. Alford and Kai Schwenzer. “What the Timing of Millisecond Pulsars Can Teach  
593 us about Their Interior”. In: *Phys. Rev. Lett.* 113.25 (2014), p. 251102. DOI: 10.1103/  
594 PhysRevLett.113.251102. arXiv: 1310.3524 [astro-ph.HE].
- 595 [9] Vardan Khachatryan et al. “Evidence for collectivity in pp collisions at the LHC”. In:  
596 *Phys. Lett.* B765 (2017), pp. 193–220. DOI: 10.1016/j.physletb.2016.12.009. arXiv:  
597 1606.06198 [nucl-ex].
- 598 [10] Jaroslav Adam et al. “Enhanced production of multi-strange hadrons in high-multiplicity  
599 proton-proton collisions”. In: *Nature Phys.* 13 (2017), pp. 535–539. DOI: 10.1038/  
600 nphys4111. arXiv: 1606.07424 [nucl-ex].
- 601 [11] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. “The Anti-k(t) jet clustering al-  
602 gorithm”. In: *JHEP* 04 (2008), p. 063. DOI: 10.1088/1126-6708/2008/04/063. arXiv:  
603 0802.1189 [hep-ph].
- 604 [12] Vardan Khachatryan et al. “Charged-particle nuclear modification factors in PbPb and  
605 pPb collisions at  $\sqrt{s_{NN}} = 5.02$  TeV”. In: *JHEP* 04 (2017), p. 039. DOI: 10.1007/  
606 JHEP04(2017)039. arXiv: 1611.01664 [nucl-ex].
- 607 [13] Betty Abelev et al. “Centrality Dependence of Charged Particle Production at Lar-  
608 ge Transverse Momentum in Pb–Pb Collisions at  $\sqrt{s_{NN}} = 2.76$  TeV”. In: *Phys. Lett.*  
609 B720 (2013), pp. 52–62. DOI: 10.1016/j.physletb.2013.01.051. arXiv: 1208.2711  
610 [hep-ex].
- 611 [14] Carlos A. Salgado and Urs Achim Wiedemann. “Calculating quenching weights”. In: *Phys.*  
612 *Rev.* D68 (2003), p. 014008. DOI: 10.1103/PhysRevD.68.014008. arXiv: hep-ph/0302184  
613 [hep-ph].
- 614 [15] Yuri L. Dokshitzer and D. E. Kharzeev. “Heavy quark colorimetry of QCD matter”. In:  
615 *Phys. Lett.* B519 (2001), pp. 199–206. DOI: 10.1016/S0370-2693(01)01130-3. arXiv:  
616 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. A* 29 (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://kndrck.co/posts/capsule\\_networks\\_explained/](https://kndrck.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).

- 707 [49] Andrew P Bradley. “The use of the area under the ROC curve in the evaluation of  
708 machine learning algorithms”. In: *Pattern recognition* 30.7 (1997), pp. 1145–1159.