Uporaba strojnega učenja za iskanje nove fizike na LHC

Simon Bukovšek

Mentor: prof. dr. Borut Paul Kerševan

Seminar, 3. letnik

Oddelek za fiziko, FMF, UL, 2023/2024

Povzetek
Strojno učenje je vedno pomembnejše orodje v fiziki visokih energij. V tem seminarju je predstavljeno delovanje detektorja ATLAS na Velikem hadronskem trkalniku, osnovni koncepti strojnega učenja in dve vrsti uporabe nevronskih mrež za odkrivanje nove fizike iz meritev eksperimenta ATLAS. Prva je uporaba klasifikatorjev za ločevanje signala od ozadja, druga pa uporaba avtoenkoderjev za iskanje morebitnih odstopanj od Standardnega modela.

Kazalo

1 Uvod **3**

2 Teoretično ozadje **3**

2.1 Nova fizika 4

3 LHC in ATLAS **4**

3.1 Sestava detektorja 5

3.2 Simulacije trkov 5

3.3 Signal in ozadje 6

4 Strojno učenje **6**

4.1 Nevronske mreže 7

4.2 Učenje nevronskih mrež 8

4.3 ROC in AUC 9

4.4 Avtoenkoderji 10

5 Uporaba strojnega učenja za odkrivanje nove fizike **11**

5.1 Zmanjševanje potrebe po uporabi high-level značilk 11

5.2 Uporaba avtoenkoderja za zaznavo odstopanj 13

6 Zaključek **14**

1 Uvod

Dandanes je v eksperimentalni fiziki osnovnih delcev eden od glavnih načinov odkrivanja nove fizike trkanje delcev v pospeševalnikih in opazovanje nastalih produktov. Večina nastalih težkih delcev razpade na manjše, še preden bi jih lahko zaznali z detektorji. Pogosto se zgodi kaskada zaporednih razpadov na več manjših produktov. Naloga fizikov je, da iz gibalnih količin in smeri končnih delcev – elektronov, mionov, protonov, nevtronov, nevtrinov in fotonov – sklepajo o osnovnih produktih trka [1]. To je podobno, kot opazovati valove na obali in iz tega sklepali, koliko in kako težke kamne je nekdo vrgel na sredino ribnika. K sreči nam pomagata dve dejstvi: trke lahko z izjemno natančnostjo simuliramo, kar nam da veliko količino podatkov za primerjavo, obenem pa je samo število trkov zelo veliko, kar nam da dobro statistično moč za sklepanje [2, 3]. Obe dejstvi pa tudi namigujeta, da bi za analizo lahko uporabili strojno učenje. Za to namreč potrebujemo veliko testnih podatkov, naučeni modeli pa so vsaj v teoriji lahko sposobni prepoznave vzorcev, ki jih ljudje ne moremo [4]. V tem seminarju si bomo ogledali, kako pridobimo podatke iz pospeševalnika, kako delujejo nevronske mreže in kako jih lahko uporabimo na pridobljenih podatkih za iskanje nove fizike.

2 Teoretično ozadje

Fiziko osnovnih delcev in njihovih interakcij opisuje *Standardni model* (SM) [5]. Ta model je zelo uspešen, saj z veliko natančnostjo napove rezultate mnogih eksperimentov, ki jih z današnjo opremo uspejo izvesti. Model opisuje vrsto osnovnih delcev, ki med seboj interagirajo preko treh osnovnih interakcij – elektromagnetne, šibke in močne. Posredniki teh interakcij so posebne vrste delci, tako imenovani *umeritveni bozoni*. Standardni model opiše vsak delec kot svoje polje, njihove interakcije pa kot sklopitve med temi polji. Delce lahko razdelimo med kvarke, leptone in bozone, kvarke in leptone pa še dodatno delimo na tri generacije. Razvrstitev osnovnih delcev po Standardnem modelu je prikazana na sliki 1.

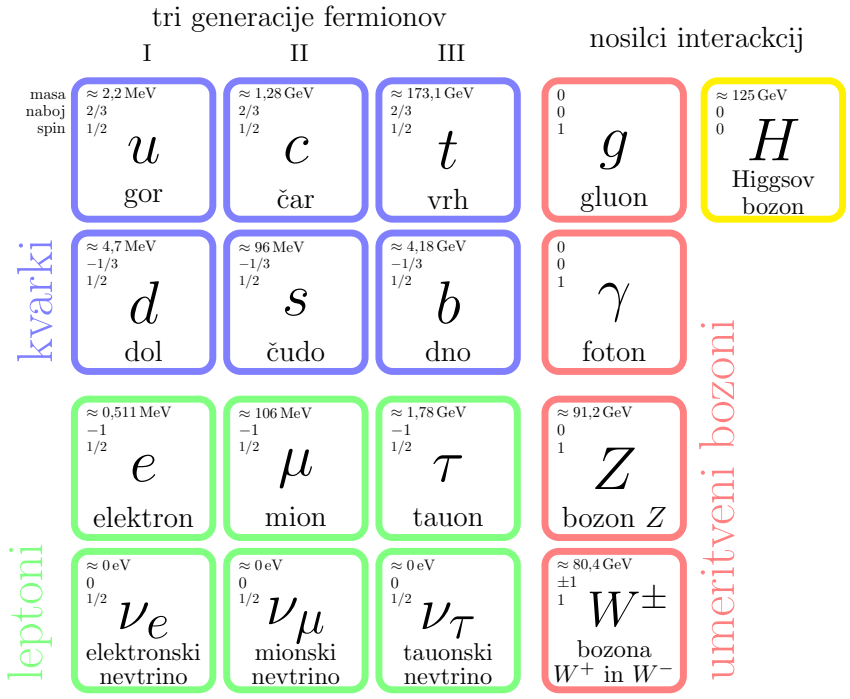
Število dogodkov N opazovanega procesa izračunamo po enačbi

$$N = \sigma \int \mathcal{L}\,\mathrm{d}t = \sigma L\,, \tag{1}$$

kjer je $\mathcal{L}$ *luminoznost* (angl. luminosity) pospeševalnika, L integrirana luminoznost, σ pa *sipalni presek* reakcije, ki ga dobimo iz teoretičnih izračunov. Lunimoznost merimo v enotah delcev na enoto časa na enoto površine in je izključno lastnost pospeševalnika, torej ni odvisna od procesa, ki ga opazujemo. Za trkanje dveh snopov delcev v okroglem pospeševalniku velja

$$\mathcal{L} = \frac{N^2 n_\mathrm{b} f}{A}\,. \tag{2}$$

Pri tem N označuje število delcev v posamezni skupini (angl. bunch), n_b število skupin, razporejenih po krogu, f frekvenco delcev po pospeševalniku in A efektiven presek snopa na točki trka [7, 5].



Slika 1: Standardni model osnovnih delcev. Vrednosti so vzete iz [6]. Mase nevtrinov so neničelne, vendar so izjemno majhne in so zato označene kot skoraj ničelne.

2.1 Nova fizika

Četudi je Standardni model najnatančnejša fizikalna teorija, kar jih današnja fizika pozna, je hkrati povsem jasno, da je nepopolna. Poleg tega da ne opisuje gravitacije, pušča neodgovorjenih nekaj pomembnih vprašanj, kot so: zakaj je vesolje sestavljeno iz materije in ne antimaterije, kaj sta temna energija in temna snov, zakaj imajo osnovni delci tako različne mase in zakaj so moči osnovnih interakcij tako različne. Posamezne probleme lahko rešimo z dodajanjem novih delcev in interakcij, kar nam odpre množico novih možnosti in teorij. Te teorije označujemo kot *onkraj Standardnega modela* (angl. beyond the Standard model, BSM). Skušamo jih potrditi tako, da v množici nastalih delcev v trkalnikih iščemo take, ki jih Standardni model ne napove [8]. Kot je že iz uvoda jasno, je to metaforično zelo podobno iskanju igle v kopici sena.

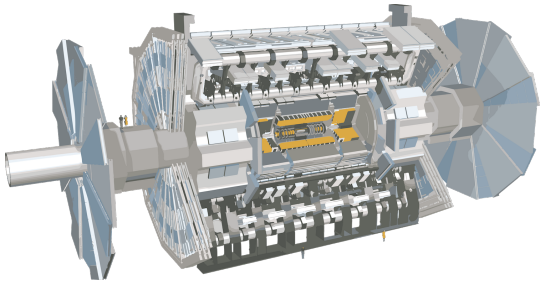
3 LHC in ATLAS

Z iskanjem novih delcev se ukvarjajo številni pospeševalniki po svetu, največji in najmočnejši med njimi je *Veliki hadronski trkalnik* (angl. Large Hadron Collider, LHC) v Evropskem centru za jedrske raziskave (fr. Conseil Européen pour la Recherche Nucléaire, CERN) v Ženevi. LHC je krožni pospeševalnik z obsegom 27 km, v katerem v ultra-visokem vakuumu pospešujejo dva snopa hadronov (tipično protonov) v nasprotni smeri do energij okrog 7 TeV. To ustreza Lorentzovemu faktorju reda 10^4. Po obsegu je razporejenih več detektorjev, kjer lahko delci iz obeh cevi med seboj trkajo. Poudarek tega seminarja je na detektorju ATLAS (angl. A Toroidal LHC ApparatuS) [9]. Protone v presledkih trkajo več let, da pridobijo kar se da veliko integrirano luminoznost in posledično čim več trkov. Med letoma 2015 in 2018 so pri invariantni masi $\sqrt{s} = 13\,\mathrm{TeV}$ pridobili integrirano luminoznost okrog $140\,\mathrm{fb}^{-1}$ [2]. Podatki, pridobljeni v tem času, se

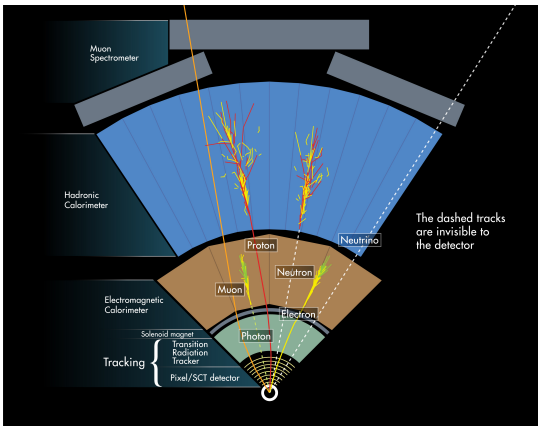
imenujejo “Run 2”. Invariantna masa je pri teh energijah kar enaka dvakratniku energije posameznega protona pri trku. Enota barn (b) ustreza površini $10^{-28}\,\mathrm{m}^2$. Po enačbi (1) lahko pričakujemo, da se je dogodek s prečnim presekom $\sigma = 1\,\mathrm{fb}$ zgodil približno 140-krat.

3.1 Sestava detektorja

Pri trku dveh protonov lahko nastane cela množica delcev. Mnogo je zelo kratkoživih in razpadejo v druge delce, še preden lahko priletijo do senzorjev. Detektor je sestavljen tako, da lahko vsaj približno ugotovi smeri in energije tistih delcev, ki jih lahko zazna. Sestavljen je iz več komponent (slika 2a), ki si čebulasto sledijo od osi trka navzven. Najbližje osi trka so sledilni detektorji, ki s pomočjo silicijevih polprevodnih ploščic zaznavajo nabite delce. Nahajajo se v močnem magnetnem polju (približno 2 T) in iz krivinskega polmera sledi delcev z natančnostjo nekaj odstotkov izračunajo njihove gibalne količine. V naslednjem sloju sta hadronski in elektromagnetni kalorimeter, ki zaznavata energijo hadronov in fotonov. V zunanjem sloju so nameščeni detektorji za zaznavanje mionov, ki so poleg nevtrinov in morebitnih eksotičnih delcev edini, ki lahko preletijo vse prejšnje sloje (glej sliko 2b). Nevtrinov se ne da enostavno zaznati, vendar se njihovo prečno energijo in azimutalno smer lahko določi kot manjkajoči del energije vseh zaznanih delcev [10].



(a) Prikaz detektorja ATLAS. Vzeto iz [1].



(b) Prečni prerez detektorja. Vzeto iz [11].

Slika 2: Detektor ATLAS. Na obeh slikah se vidi čebulna struktura, ki je ključna za zaznavanje delcev pri trku.

Skupine bližnjih delcev se s pomočjo algoritmov združi v *pljuske* (angl. jets), ki nam dajo bolj neposredne informacije o nastalih delcih takoj po trku. Pljuskom se lahko določi smer in energijo, z algoritmi strojnega učenja pa lahko pa tudi ali so nastali iz kvarka b ali c (tako imenovana b-označevanje in c-označevanje, angl. tagging) [12].

3.2 Simulacije trkov

Rekonstrukcija dogodkov iz končnih razpadnih produktov je izjemno zahtevna naloga. Del rešitve je uporaba simulacij trkov. S tem lahko rekonstruirajo posamezne pljuske in

s tem zmanjšajo količino relevantnih podatkov za posamezen dogodek, obenem pa simulacije lahko služijo za treniranje algoritmov strojnega učenja. Večino simulacij temelji na metodi Monte Carlo (MC), ki s pomočjo žrebanja naključnih številk simulira razpadne procese. Velika ovira predstavlja dejstvo, da protoni niso osnovni delci, in da se jih ne da perturbativno izračunati iz kvantne teorije polja. Veliko teh algoritmov tako temelji na rezultatih preteklih eksperimentov in k sreči se najnaprednejši algoritmi dobro ujemajo med sabo. Po simulaciji trkov, ki izračunajo vektorje četverce gibalnih količin vseh nastalih delcev, je potrebno simulirati tudi interakcijo med delci in detektorjem. S tem se poustvari dejanska meritev, ki jo dobijo iz detektorja. Na koncu lahko dodajo še šum in trke prekrijejo enega čez drugega ter s tem lahko sklepajo o sestavi dejanskih meritev. Prednost simulacij je tudi ta, da lahko oblikujejo hipotetične odzive na procese, ki vsebuje kake neodkrite delce, denimo supersimetrične delce, kvarke četrte generacije, nove umeritvene bozone, ali pa celo procese, ki se odvijajo v višjih dimenzijah [3, 13].

Kljub temu da lahko s pomočjo simulacij na podlagi končnih meritev sklepamo o sestavnih delih prvotne reakcije, je to še vedno zapleten proces. V zadnjih letih je razvoj računalnikov in grafičnih kartic omogočil, da lahko del tega sklepanja prepustimo samoučečim računalniškim algoritmom.

3.3 Signal in ozadje

Ko iščemo nove procese v trkalniku, je naša glavna naloga, da jih razločimo od kopice drugih procesov, ki dajo zelo podobne razpadne produkte. Iskan proces, oziroma njegove produkte, imenujemo *signal*, vse ostale že poznane procese pa *ozadje*. Novi delci lahko razpadejo na več različnih načinov, imenovanih *razpadni kanali*, na vsakega s svojo verjetnostjo. Pri preučevanju novega delca se običajno omejimo na iskanje samo enega razpadnega kanala. Pri analizi želimo, da je verjetnost za ta razpad čim večja, obenem pa tudi, da se vzporedno ne dogaja še kak drug, že znan proces z enakimi razpadnimi produkti in podobno energijo. Ločevanje signala od ozadja je torej ključnega pomena, hkrati pa je to nekaj, kar se lahko računalniki z nekaj zunanje pomoči naučijo sami [12].

4 Strojno učenje

Strojno učenje je široko področje računalništva in matematike in zanj ne obstaja enotna definicija. V kontekstu tega seminarja je nemara najbolj primerno, če si izposodim izjavo Arthurja Samuela iz leta 1959: "Strojno učenje je področje, ki se ukvarja z omogočanjem računalnikov, da se učijo, ne da bi jih za to eksplicitno programirali." To zavzema vse od najenostavnejše linearne regresije do najmodernejših generativnih jezikovnih modelov. V tem poglavju si bomo ogledali nekaj osnovnih konceptov strojnega učenja, v nadaljevanju pa se bomo posvetili dvema konkretnima realizacijama, ki nam lahko pomagata pri odkrivanju nove fizike: klasifikatorji ter avtoenkoderji.

Cilj strojnega učenja je, da iz nabora vhodnih podatkov, imenovanih *značilke* (angl. features), napove izhodne podatke, imenovane *oznake* (angl. labels). Značilke so lahko kakršnikoli številski podatki, denimo svetlosti slikovnih točk na sliki, neka poved, ki jo vnesemo v jezikovni model, ali pa gibalne količine in smeri pljuskov v detektorju. Oznake

nam lahko predstavljajo različne stvari – pri klasifikaciji so to različne kategorije, recimo tip oblačila na sliki, lahko je to verjetnost za posamezni dogodek, ali pa je celotno besedilo, ki ga vrne jezikovni model [4].

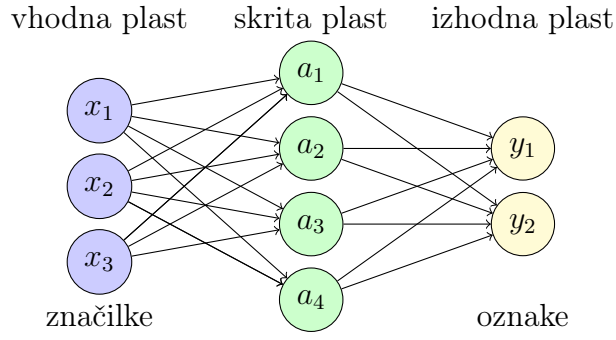
Preden pa je računalnik lahko sposoben smiselno tvoriti oznake, ga je potrebno to naučiti. V preprostem modelu podajamo sistemu značilke skupaj s pripadajočimi oznakami. Sistem nato preveri, kakšno oznako bi dal podanim značilkam in nato popravi svoje parametre tako, da se čim bolj približa pravilnim oznakam. Tako začne model sam povezovati določene lastnosti značilk, ne da bi mu to bilo eksplicitno povedano. Večji kot je model in več kot ima parametrov, bolj zapletene povezave lahko poišče in vrne natančnejše rezultate. Seveda pa z večanjem modela naletimo na druge težave, kot so velika prostorska in časovna zahtevnost. Drug način je, da sistemu podajamo samo ene vrste podatke, recimo samo procese ozadja. Tako se sistem nauči, kaj naj pričakuje, in vsako odstopanje ustvari nepričakovan odziv. Tak pristop imenujemo *zaznava odstopanj* (angl. anomaly detection).

Pomembno se je zavedati, da različni modeli strojnega učenja delujejo najbolje na različnih tipih podatkov in pogosto je na danih podatkih potrebno preizkusiti več različnih modelov, da se najde najprimernejši [4]. V fizikalnih uporabah strojnega učenja lahko močno koristi, da značilke odražajo fizikalne lastnosti – denimo, da poleg gibalne količine curka podamo še njen kvadrat, ki je Lorentzovo invarianten. Prav tako je pomembno, da se številske vrednosti zančilk ne raztezajo preko več redov velikosti, zato se fizikalne podatke pogosto normalizira, preden se jih poda modelu.

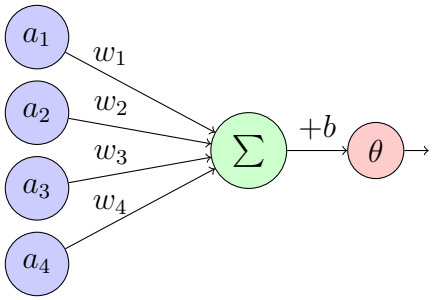
4.1 Nevronske mreže

Nevronska mreža (angl. neural network, NN) je računalniški model, ki posnema biološke mreže nevronov. Sestavljena je iz več plasti nevronov. Začetna plast se imenuje *vhodna* in predstavlja značilke modela, zadnja pa se imenuje *izhodna* in predstavlja oznake. Vmes se nahajajo *skrite plasti* (glej sliko 3a). Če je teh plasti več, se nevronski mreži reče *globoka* (DNN). Posamezen nevron je povezan z vsemi nevroni v prejšnji plasti, pri čemer vsaka povezava predstavlja neko utež. Značilke aktivirajo vhodni sloj nevronov, ti pa preko povezav z naslednjimi sloji na utežen način prenesejo aktivacijo na nadaljnje sloje. Končna aktivacija izhodnega sloja je napoved modela. Med fazo učenja se uteži modela po majhnih korakih spreminjajo.

Naj vektor $\boldsymbol{x} \in \mathbb{R}^n$ predstavlja značilke in vektor $\boldsymbol{a} \in \mathbb{R}^m$ aktivacije naslednjega sloja tako kot na sliki 3a. Naj bo velikost vektorja $\boldsymbol{x}$ enaka n, velikost vektorja $\boldsymbol{a}$ pa enaka m. Število puščic, ki jih lahko narišemo med slojema, in hkrati število uteži je enako $n \cdot m$, zato lahko vse uteži zapišemo v obliki matrike $\mathsf{W} \in \mathbb{R}^{m \times n}$. Vsak nevron v skriti plasti se aktivira z uteženim seštevkom aktivacij prejšnjega sloja, zraven pa se doda še *statistična pristranskost* (angl. bias), ki ga označimo z b. To je le dodaten parameter nevronske mreže, ki se spreminja z učenjem. S tem smo dosegli linearno kombinacijo značilk – skupaj s pristranskostjo b matematiki bi temu rekli afina transformacija. Ker pa s kopičenjem takih transformacij ne moremo modelirati zapletenih vzorcev, moramo na koncu vsako aktivacijo še prenesti skozi neko nelinearno funkcijo θ, ki ji rečemo *aktivacijska funkcija*. Ta funkcija mora imeti dobro definiran odvod in biti enostavna za računanje. Najpogosteje



(a) Primer enostavne nevronske mreže. Mreža vsebuje tri značilke in dve oznaki ter ima en skriti sloj z dimenzijo 4. Vsaka puščica predstavlja utež. Značilke lahko zberemo v vektor $\boldsymbol{x} = (x_1, x_2, x_3)$, oznake pa v vektor $\boldsymbol{y} = (y_1, y_2)$.



(b) Prikaz delovanja enega nevrona. Vsaka povezava predstavlja utež w, utežene aktivacije prejšnjega sloja se seštejejo, doda se pristranskost b in se nato prenese skozi aktivacijsko funkcijo θ.

Slika 3: Dva prikaza komponent nevronske mreže.

se uporabljajo logistična funkcija[1] in *usmerjena linearna enota*[2] (angl. rectified linear unit, ReLU). Za posamezen nevron velja

$$a_i = \theta(x_1 w_{i1} + x_2 w_{i2} + \cdots + x_n w_{in} + b_i) \,, \tag{3}$$

skupaj pa lahko aktivacijo celotnega sloja zapišemo z vektorsko enačbo

$$\boldsymbol{a} = \boldsymbol{\theta}(\mathsf{W}\boldsymbol{x} + \boldsymbol{b}) \,, \tag{4}$$

kjer je $\boldsymbol{b} \in \mathbb{R}^m$ vektor pristranskosti. Tako deluje tudi vsaka naslednja plast, pri čemer se vektor $\boldsymbol{a}$ iz prejšnje plasti uporabi kot vektor $\boldsymbol{x}$ za naslednjo plast. Aktivacijske funkcije se lahko med sloji razlikujejo, zlasti pri izhodnem sloju, kjer pogosto želimo da so oznake normirane na interval $[0, 1]$. Velikost in število skritih plasti in vrste aktivacijskih funkcij so lastnosti modela, ki jih določimo pred začetkom učenja. Pogosto je potrebno preizkusiti več različnih konfiguracij, da lahko najdemo najboljšo [4].

4.2 Učenje nevronskih mrež

Preden lahko z modelom napovedujemo oznake, ga je potrebno naučiti. V primeru klasifikatorja uporabljamo nadzorovano učenje, kjer modelu ob značilkah podamo tudi pravilne oznake. Učenje predstavlja algoritem *optimizator*, ki na podlagi napake napovedi modela popravi uteži in pristranskosti. Najprej je potrebno definirati neko *funkcijo izgube* L (angl. loss function), ki meri razliko med napovedmi modela in pravilnimi oznakami. Najenostavnejša taka funkcija je povprečen kvadrat napake (MSE), ki jo zapišemo kot

$$L_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \,, \tag{5}$$

[1] Logistična funkcija ali *sigmoida* ima predpis $\sigma(x) = 1/(1 + e^{-x})$.
[2] $\text{ReLU}(x) = \max\{x, 0\}$.

kjer smo z $\hat{y}_i$ označili pravilne oznake, z y_i pa napovedi modela. Modeli se bolje učijo, če uteži ne popravljamo po vsakem naboru značilk posebej, ampak pošljemo več primerov in delamo popravke glede na povprečje napak. Taka skupina se imenuje *skupek* (angl. batch), njena velikost N pa nastopa v enačbi (5). Cilj optimizacijskega algoritma je poiskati minimum funkcije izgube, ki je odvisen od uteži W in pristranskosti $\boldsymbol{b}$. Prosti parametri modela se lahko merijo v milijonih, funkcija izgube pa se nahaja v prav toliko dimenzijah. Najenostavnejši optimizator je *vzvratna propagacija napake* (angl. backpropagation), ki išče minimum s pomočjo majhnih korakov v smeri gradienta funkcije izgube. Ker so nevronske mreže praktično linearne transformacije, opremljene z nelinearnimi aktivacijskimi funkcijami, je gradient funkcije izgube glede na uteži mogoče izračunati z verižnim pravilom odvajanja. To je tudi razlog, zakaj so aktivacijske funkcije izbrane tako, da so njihovi odvodi enostavni. Posamezen korak optimizacije lahko zapišemo kot

$$W_{\text{novi}} = W_{\text{stari}} - \alpha \nabla_W L \,, \tag{6}$$

kjer W predstavlja nabor vseh uteži in pristranskosti v celotni nevronski mreži, α pa je *korak učenja* (angl. learning rate). Ta določa, kako velik korak naredimo v smeri gradienta. Na začetku je smiselno imeti dovolj velik korak, da se model hitro približa minimumu, kasneje pa je potrebno korak zmanjševati, da se model lahko ustali [4]. Obstajajo tudi naprednejši in učinkovitejši optimizatorji; dandanes je med najpopularnejšimi *Adam* [14].

Testne podatke običajno ločimo na dva dela: podatke za učenje in podatke za preverbo. Model učimo na podatkih za učenje, občasno pa preverimo, kako dobro model napove oznake podatkov za preverbo, ki jih model še ni videl. S tem se izognemo *pretreniranju* (angl. overtraining), ki se zgodi, ko se model preveč specificira na testne podatke.

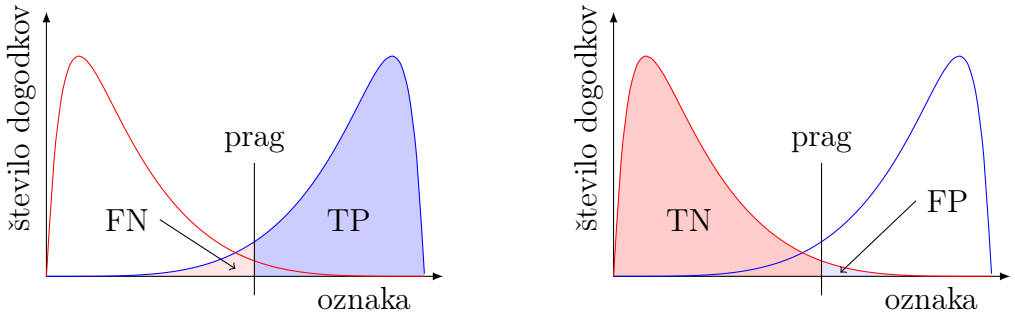
4.3 ROC in AUC

Ko končamo z učenjem nevronske mreže, potrebujemo neko mero uspešnosti modela. Za lažjo predstavo privzemimo, da mreža vrne samo eno oznako, ki je zvezna spremenljivka na intervalu $[0, 1]$. Denimo, da to v konkretnem primeru pomeni prisotnost Higgsovega bozona v reakciji. Ker je to zvezna spremenljivka, lahko izberemo poljuben prag (angl. treshold), ki ločuje prisotnost in odsotnost bozona. Ko si izberemo neko vrednost praga, dobimo štiri vrste primerov (glej sliko 4):

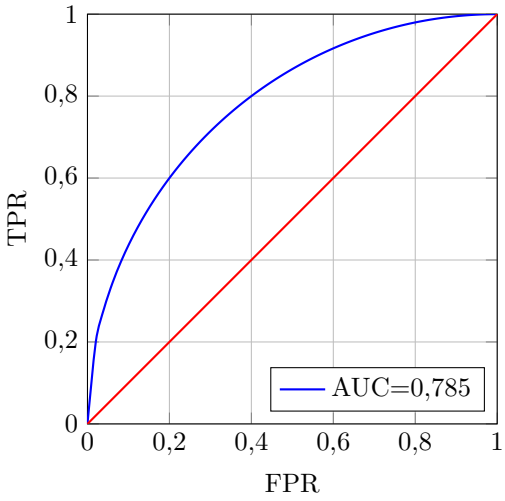
- *pravilno pozitivni* (TP): model je pravilno napovedal prisotnost bozona,
- *lažno pozitivni* (FP): model je napovedal prisotnost bozona, v resnici pa ga ni bilo,
- *pravilno negativni* (TN): model je pravilno napovedal odsotnost bozona,
- *lažno negativni* (FN): model je napovedal odsotnost bozona, v resnici pa je bil.

Pri določanju uspešnosti modela nas predvsem zanima kako so te štiri količine odvisne od izbire meje. V ta namen vpeljemo dve novi količini, imenovani *delež pravilno pozitivnih primerov* (angl. true positive rate, TPR) in *delež lažno pozitivnih primerov* (angl. false positive rate, FPR). Definiramo ju kot

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \,, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \,. \tag{7}$$



Slika 4: Prikaz pravilno pozitivnih in negativnih ter lažno pozitivnih in negativnih primerov. Rdeča krivulja predstavlja porazdelitev negativnih primerov po oznaki, modra pa pozitivnih. Pokončna črna črta označuje izbran prag.

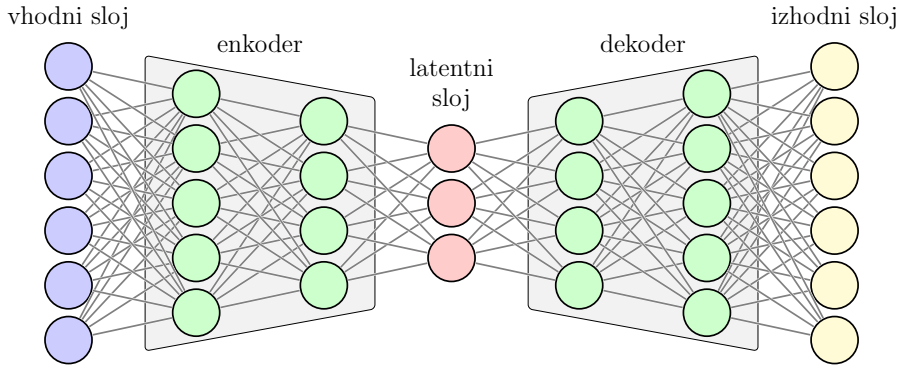


Slika 5: Primer ROC krivulje. Modra krivulja predstavlja natreniran model, rdeča pa netreniran. Ploščina pod krivuljo je AUC. Boljši kot je model, bolj se krivulja približa levemu zgornjemu kotu in večji je AUC.

TPR predstavlja delež dejanskih pozitivnih primerov med vsemi, ki jih je model napovedal kot pozitivne. FPR pa predstavlja delež dejanskih negativnih primerov med vsemi, ki jih je model napovedal kot negativne. Obe količini sta odvisni od izbire praga, zato narišemo parametrično krivuljo obeh količin, ko prag preteče vse vrednosti med najnižjo in najvišjo. Nastali krivulji rečemo ROC (angl. reciever operating characteristic). Popoln model, ki bi znal pravilo klasificirati vse primere, bi imel ROC krivuljo, ki bi šla skozi točko $(0, 1)$ – Heavisideova funkcija. Popolnoma nakljućen model bi imel ROC krivuljo, ki bi bila enaka premici $y = x$ (rdeča krivulja na sliki 5). Bolj kot se ROC krivulja približa levemu zgornjemu kotu, boljši je model. Za enostavno primerjavo med modeli lahko izračunamo površino pod ROC krivuljo (AUC, angl. area under curve). Ta mera se zelo pogosto uporablja za primerjave med različnimi modeli. Zavzame lahko vrednosti med $1/2$ in 1 – večja kot je vrednost, boljši je model [4].

4.4 Avtoenkoderji

Avtoenkoderji so posebna vrsta nevronskih mrež, primarno namenjeni zmanjševanju dimenzije podatkov. Omogočajo nam nenadzorovano učenje, kar ima v fizikalnem kontekstu nekaj prednosti. V osnovni različici je avtoenkoder nevronska mreža, kjer skrite sloje lahko razdelimo v tri dele. Prvi del je *enkoder*, ki ga sestavljajo postopoma manjši sloji nevro-



Slika 6: Shematska sestava preprostega avtoenkoderja. Vhodni in izhodni sloj imata šest parametrov, latentni sloj pa samo tri.

nov. Drugi del je *latentni sloj*, ki ima manj nevronov kot vhodni sloj in tako z manj parametri predstavlja vhodne podatke. Tretji del je *dekoder*, ki ga sestavljajo postopoma večji sloji. Tipično je dekoder kar zrcalna slika enkoderja (glej sliko 6). V avtoenkodejih ima izhodni sloj vedno enako dimenzijo kot vhodni.

Uspešnost celotne mreže merimo kot razliko med vhodnimi in izhodnimi podatki. Cilj je, da se model nauči čim bolje poustvariti vhodne podatke, pri čemer jih mora vmes zmanjšati na manjše število dimenzij. S tem se nauči izluščiti najpomembnejše lastnosti podatkov.

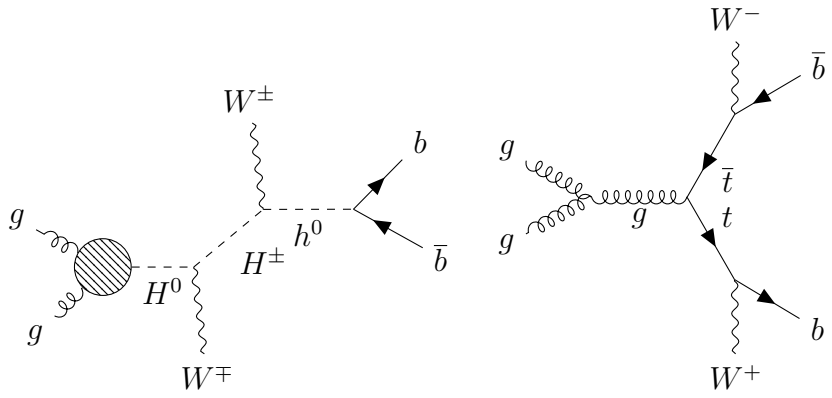
Če se tak model uči samo na eni vrsti podatkov – recimo na procesih ozadja v pospeševalniku – in se jih nauči dobro rekonstruirati, potem tega morda ne zna dobro storiti za procese signala. Če torej v naučen model podamo nek dogodek in model vrne približno iste oznake nazaj, lahko sklepamo, da je dogodek podoben tistim, na katerih se je model učil – torej ozadju. Če pa model vrne bistveno drugačne oznake, je to lahko znak, da je dogodek nekaj posebnega – signal. To je en primer implementacije zaznave odstopanj. Pri trkih v eksperimentu ATLAS nam tak pristop lahko koristi, saj se z njim ne zavežemo posebni vrsti BSM teorije, ampak lahko zaznavamo odstopanja od SM v kateremkoli (tudi še neteoretiziranem) procesu [4].

5 Uporaba strojnega učenja za odkrivanje nove fizike

Strojno učenje dandanes sili že v vsak korak v procesu obdelave podatkov iz ATLAS detektorja, od simulacij dogodkov do združevanja razpadnih produktov v pljuske, b-označevanje in odkrivanje novih delcev [15, 16, 17, 18]. V tem razdelku se bomo posvetili dvema primeroma uporabe strojnega učenja za iskanje novih procesov iz meritev detektorja ATLAS.

5.1 Zmanjševanje potrebe po uporabi high-level značilk

Osredotočimo se na hipotetičen razpad težkega nevtralnega Higgsovega bozona H_0 preko hipotetičnega nabitega težkega Higgsovega bozona $H^{\pm}$ in potrjenega lahkega Higgsovega bozona h_0 na dva W bozona in dva b kvarka. Razpadni proces je prikazan na sliki 7a. Ta proces je v tem primeru signal. Proces ozadja, ki se najbolj prekriva z iskanim razpadom, je prikazan na Sliki 7b. Začetni in končni produkti so v tem primeru popolnoma enaki, edina razlika je kinematiki: signal bo imel resonančni vrh pri masi H_0, ozadje pa ne. W



(a) Signal. **(b)** Ozadje.

Slika 7: Feynmanova diagrama za razpad težkega Higgsovega bozona H^0 na $W^+W^-b\bar{b}$ (signal) in razpad gluona na iste produkte (ozadje). Šrafirana pika predstavlja nepoznan in nepomemben del procesa.

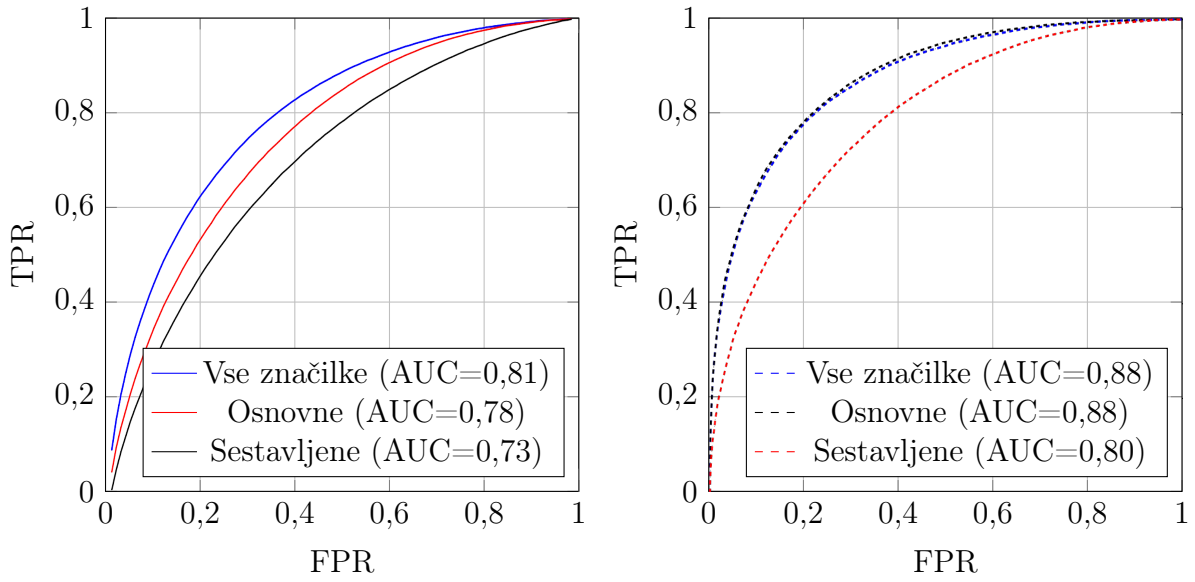
	AUC		
Model	Osnonve	Sestavljene	Vse značilke
NN	$0{,}733 \pm 0{,}007$	$0{,}777 \pm 0{,}001$	$0{,}816 \pm 0{,}004$
DN	$0{,}880 \pm 0{,}001$	$0{,}800 \pm 0{,}001$	$0{,}885 \pm 0{,}002$

Tabela 1: Vrednosti AUC pri uporabi različnih vrst značilk za navadne (NN) in globoke (DN) nevronske mreže [12]

bozoni tipično razpadejo na lepton (ℓ) in pripadajoči nevtrino (ν) ali pa dva kvarka, ki tvorita dva pljuska delcev (jj).

S pomočjo prej naštetih orodij (glej podpoglavje 3.2) so simulirali okorg pol milijona primerov signala in ozadja pri $m_{H^0} = 425\,\text{GeV}$ in $m_{H^\pm} = 325\,\text{GeV}$. Za vsak primer razpada je simulacija priskrbela 21 osnovnih (angl. low-level) in 7 sestavljenih (angl. high-level) značilk [19]. Osnovne značilke predstavljajo velikosti in smeri gibalnih količin štirih pljuskov (skupaj 12 značilk), njihove b-oznake (4 značilke), smer in gibalno količino nastalega leptona (3 značilke) ter smer nastalega nevtrina (2 značilki). Sestavljene značilke so preračunane iz osnovnih in predstavljajo skupno invariantno maso nekaterih komponent, kot je denimo skupna invariantna masa nevtrina in leptona $m_{\ell\nu}$, ki bi morala imeti resonanco pri masi W bozona. To so zelo relevantne fizikalne količine, ki močno pospešijo učenje plitvih nevronskih mrež. Kaj pa če obstajajo katere pomembne izpeljane količine, za katere morda še ne vemo? Bilo bi zelo koristno, če bi model znal sam izluščiti najpomembnejše lastnosti podatkov. V Ref. [12] so se lotili prav tega. Preverili so, kako dobro se na posameznih značilkah uči "navadna" nevronska mreža (NN) z enim samim skritim slojem in globoka nevronska mreža (DN) z do 6 skritimi sloji. Primerjali so vrednosti AUC pri različnih modelih, rezultati so prikazani v tabeli 1, krivulje ROC pa na slikah 8a in 8b.

Opazimo lahko, da se globoka nevronska mreža ne nauči nič kaj bolje, če ji poleg osnovnih značilk podamo še izračunane sestavljene spremenljivke. Iz tega lahko sklepamo, da se jih je naučila sama. Pri navadni nevronski mreži pa dodane sestavljene značilke vidno pripomorejo k izboljšanju vrednosti AUC. V splošnem globoka nevronska mreža v vseh kategorijah prehiti navadno. Slabost takega pristopa se odraža v tem, da nam globoka nevronska mreža ne poda nobenega fizikalnega vpogleda v morebitne sestavljene značilke, saj celotna mreža deluje kot črna škatla.

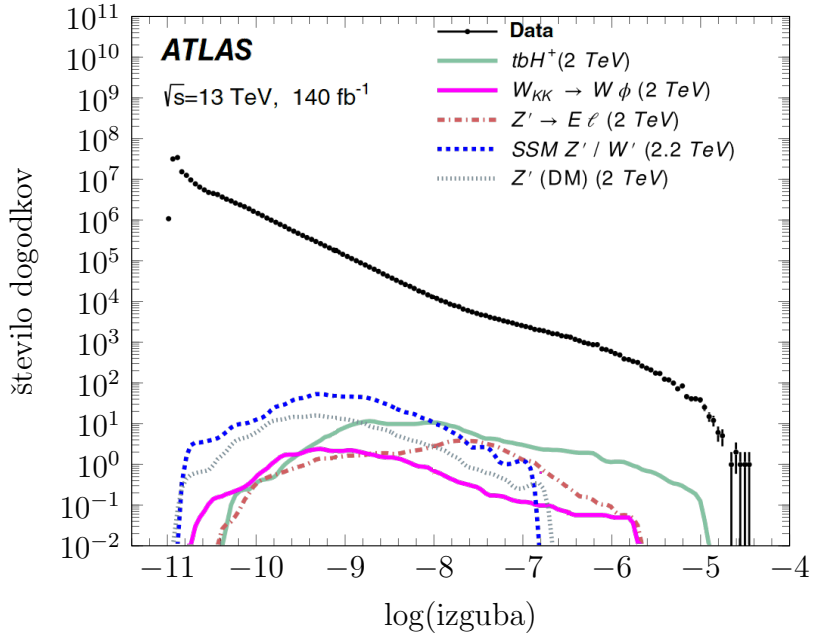


(a) Krivulja ROC za NN. **(b)** Krivulja ROC za DN.

Slika 8: ROC krivulji za navadno (NN) in globoko (DN) nevronsko mrežo pri različnih vrstah značilk. Povzeto iz [12].

5.2 Uporaba avtoenkoderja za zaznavo odstopanj

Drug pristop k odkrivanju nove fizike je zaznava odstopanj. Teoretično delovanje avtoenkoderjev je opisano v podpoglavju 4.4, tukaj pa si bomo pogledali eno od realizacij. V Ref. [18] so vzeli 1278 značilk, ki poleg energij in smeri do 10 različnih pljuskov in več leptonov vsebuje še izračunane kombinacije invariantnih mas vseh možnih parov med nastalimi pljuski in leptoni. Nevronsko mrežo so sestavili iz skritih slojev z dimenzijami 800, 400, 200, 400, 800 v tem vrstnem redu, izstopni sloj pa je imel prav tako dimenzijo 1278. To pomeni, da je moral sistem iz 1278 značilk v vhodnem sloju izluščiti 200 parametrov, kolikor je dimenzija latentnega sloja, iz katerih je nato skušal poustvariti vhodne podatke. Za funkcijo izgube so uporabili logaritem vsote kvadratov razlik vhodnega in izhodnega sloja. Trenirali so na simuliranih podatkih, model pa so preverili pravih meritvah iz detektorja ATLAS (Run 2). Uporabili so le tiste dogodke, kjer je nastal dovolj energetski pljusk in vsaj en dovolj energetski lepton. Ustreznost modela so preverili s simuliranimi dogodki z nekaj prevladujočih BSM teorij. Na sliki 9 je prikazana porazdelitev funkcije izgube za 166 055 597 trkov, izmerjenih na eksperimentu ATLAS (črno). Razvidno je, da je dogodkov z veliko vrednostjo funkcije izgube(več od -6) zelo malo. V barvah so prikazane porazdelitve funkcije izgube za simulirane dogodke nekaterih BSM teorij. Tem teorijam pogostost dogodkov ne pada tako hitro z večanjem izgube kot pri izmerjenih podatkih. Na žalost je samo število BSM dogodkov tako majhno, da rezultati niso prinesli nobenih signifikantnih odstopanj [18].



Slika 9: Prikaz porazdelitve funkcije izgube na 166 055 597 podatkih iz detektorja ATLAS (črno). Na isti graf so narisane porazdelitve funkcije izgube za simulacije podobnih dogodkov nekaterih prevladujočih BSM teorij (barvno), pri katerih prav tako nastane energetski lepton. Podrobnosti presegajo obseg tega seminarja. Povzeto iz [18].

6 Zaključek

Uporaba strojnega učenja vstopa z različnih koncev v tok obdelave podatkov z detektorja ATLAS. Razvoj računalniške opreme je v zadnjem desetletju omogočil hitrejšo uporabo strojnega učenja, kar je mnoge pritegnilo, da si z njim pomagajo pri odkrivanju procesov znotraj detektorja. V tem seminarju smo si ogledali splošno delovanje nevronskih mrež in njihovo uporabo pri klasificiranju posebnega razpada hipotetičnega težkega Higgsovega bozona. Globoke nevronske mreže so sposobne same določiti najpomembnejše lastnosti podatkov in jim za uspešno delovanje ni potrebno dodajati izračunanih sestavljenih značilk. Ker večina osnovnih nevronskih mrež temelji na poznavanju alternativne fizikalne teorije, ki jo poskuša potrditi ali ovreči, so nekateri avtorji poskusili uporabiti zaznavo odstopanj, ki so neodvisna od teorije BSM. Takšni pristopi za zdaj še niso uspeli dokazati signifikantnih odstopanj od Standardnega modela. Nemara ljudje še niso preizkusili pravega pristopa, morda so trenutne energije trkov premajhne, ali pa preprosto še ni bilo izmerjenih dovolj trkov, da bi na podlagi izjemno redkih dogodkov lahko z dovoljšnjo gotovostjo zaznali novo fiziko.

Literatura

[1] *Detector & Technology*. CERN. 2024. URL: `https://atlas.cern/Discover/Detector` (pridobljeno 15. 3. 2024).

[2] ATLAS Collaboration. »Luminosity determination in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector at the LHC«. V: *The European Physical Journal C* 982.83 (2023). DOI: `https://doi.org/10.48550/arXiv.2212.09379`.

[3] *Monte Carlo production chain*. ATLAS Open Data Documentation. 2024. URL: `https://opendata.atlas.cern/docs/documentation/monte_carlo/MC_production_chain/` (pridobljeno 15. 3. 2024).

[4] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Ur. R. Roumeliotis in N. Tache. 2. izd. O'Reilly Media, Inc., 2019. ISBN: 9781492032649.

[5] D. J. Griffiths. *Introduction to elementary particles; 2nd rev. version*. Physics textbook. New York, NY: Wiley, 2008. ISBN: 9783527406012. URL: https://cds.cern.ch/record/111880.

[6] Particle Data Group in sod. »Review of Particle Physics«. V: *Progress of Theoretical and Experimental Physics* 2022.8 (avg. 2022), str. 083C01. ISSN: 2050-3911. DOI: 10.1093/ptep/ptac097. eprint: https://academic.oup.com/ptep/article-pdf/2022/8/083C01/49175558/ptac097_19_miscellaneous_searches_and_searches_in_other_sections.pdf. URL: https://doi.org/10.1093/ptep/ptac097.

[7] B. Povh in sod. *Particles and Nuclei: An Introduction to the Physical Concepts*. Springer Berlin Heidelberg, 2008. ISBN: 9783540793687. URL: https://books.google.si/books?id=XyW97WGyVbkC.

[8] B. C. Allanach. »Beyond the Standard Model«. V: *CERN Yellow Rep. School Proc.* 6 (2019), str. 113–144. DOI: 10.23730/CYRSP-2019-006.113. URL: https://cds.cern.ch/record/2702850.

[9] *The Large Hadron Collider*. CERN. 2024. URL: https://home.cern/science/accelerators/large-hadron-collider (pridobljeno 15. 3. 2024).

[10] I. Efthymiopoulos. *Overview of the ATLAS detector at LHC*. Teh. poročilo 7. version revised com-conf-99-002_version1 on Tue Jun 22 21:18:36 MET DST 1999. Geneva: CERN, 1999. URL: https://cds.cern.ch/record/409257.

[11] J. Pequenao in P. Schaffner. »How ATLAS detects particles: diagram of particle paths in the detector«. 2013. URL: https://cds.cern.ch/record/1505342 (pridobljeno 24. 3. 2024).

[12] P. Baldi, P. Sadowski in D. Whiteson. »Searching for exotic particles in high-energy physics with deep learning«. V: *Nature Communications* 5.1 (jul. 2014). ISSN: 2041-1723. DOI: 10.1038/ncomms5308. URL: http://dx.doi.org/10.1038/ncomms5308.

[13] C. Bierlich in sod. *A comprehensive guide to the physics and usage of PYTHIA 8.3*. Stran 46. 2022. arXiv: 2203.11601 [hep-ph].

[14] D. P. Kingma in J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[15] S. Otten in sod. *Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer*. 2021. arXiv: 1901.00875 [hep-ph].

[16] ATLAS Collaboration in Irinel Caprini. »A neural network clustering algorithm for the ATLAS silicon pixel detector«. V: *Journal of Instrumentation* 9 (jun. 2014). DOI: 10.1088/1748-0221/9/09/P09009.

[17] Michela Paganini. *Machine Learning Algorithms for b-Jet Tagging at the ATLAS Experiment*. 2017. arXiv: 1711.08811 [hep-ex].

[18] »Search for New Phenomena in Two-Body Invariant Mass Distributions Using Unsupervised Machine Learning for Anomaly Detection at $\sqrt{s} = 13$ TeV with the ATLAS Detector«. V: *Phys. Rev. Lett.* 132 (8 feb. 2024), str. 081801. DOI: 10.1103/PhysRevLett.132.081801. URL: https://link.aps.org/doi/10.1103/PhysRevLett.132.081801.

[19] D. Whiteson. *HIGGS*. UCI Machine Learning Repository. 2014. DOI: https://doi.org/10.24432/C5V312.