

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»

УДК 539.12.01

ОТЧЕТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

**ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ
ИДЕНТИФИКАЦИИ СТРУЙ, ОБРАЗОВАННЫХ W-БОЗОНОМ**

Научный руководитель

к.ф.-м.н.

Студент

_____ Мягков Алексей Григорьевич

_____ Ван Алина Маошэновна

Москва
2023

Содержание

Введение	3
1 Машинное обучение с учителем	4
1.1 Бинарная классификация	5
1.2 BDT	5
1.3 Нейронные сети	7
2 Дискриминирующие переменные	8
3 Процесс работы и результаты	9
3.1 Построение сигнального и фонового деревьев	9
3.2 Обучение и тестирование моделей	10
3.3 Резюме результатов моделей	11
4 Заключение	14

Введение

Стандартная модель – это современная теория в физике элементарных частиц, описывающая сильное, слабое и электромагнитное взаимодействия. Однако, несмотря на все свои преимущества, Стандартная модель не считается полной теорией всего, так как она не включает в себя гравитационное взаимодействие и не описывает некоторые экспериментальные факты, такие как скрытая масса, темная энергия, барионная асимметрия и т.д. Предполагается, что Стандартная модель является частью более общей теории. Поэтому одной из главных задач на ЛНС является поиск новой физики за рамками Стандартной модели.

Идентификация толстых струй, образованных W -бозонами, распавшимися по адронной моде, является одним из главных составляющих этапов анализа данных с экспериментов по поиску новой физики. К примеру, в эксперименте по поиску возбужденного лептона с конечным состоянием $e\nu J$ процессом, приносящем основной вклад в фон, является образование пары топ-анти топ. Так как топ-кварк распадается преимущественно по слабому взаимодействию, идентификация струй, образованных W -бозоном, является необходимой для подавления фона.

В предыдущей работе для идентификации W -бозона использовалось ограничение по переменной D_2 . Однако для выделения струй от векторных бозонов существуют и другие дискриминирующие переменные. Для улучшения точности идентификации толстой струи рассмотрено решение этой задачи с помощью методов машинного обучения.

В контексте машинного обучения задача идентификации частицы – это задача бинарной классификации, т.е. предсказание по полученным признакам к какому из двух классов принадлежит объект: фону или сигналу.

Цель работы.

Целью работы является использование методов машинного обучения для решения задачи идентификации толстых струй, образованных W -бозоном.

В соответствии с поставленной целью задачами данной работы были:

- Ознакомление с основными алгоритмами машинного обучения для решения задачи бинарной классификации;
- Выбор дискриминирующих переменных (признаков) для объектов;
- Формирование сигнального и фонового деревьев для данной задачи;
- Обучение и тестирование моделей;
- Резюме результатов.

1 Машинное обучение с учителем

Машинное обучение – это область прикладной математики, изучающая методы решения задач с использованием обучающих данных. В данной работе используются методы машинного обучения с учителем.

Постановка задачи для обучения с учителем следующая: на вход алгоритма подаются размеченные данные вида (матрица признаков; отклик). Эти данные называются обучающей выборкой, они используются для настройки модели. В зависимости от задачи данные также могут разделить и на вариационную выборку, которая необходима для оценки обобщающих способностей обученных моделей и выбора лучшей модели. Тестовая выборка используется для финальной оценки качества модели. Цель машинного обучения с учителем состоит не в моделировании данных из пространства признаков, а в предсказании откликов на новых данных.

Оценка качества модели связана с функцией потерь. Функция потерь – это мера неточности модели, она сравнивает цели и выходы модели.

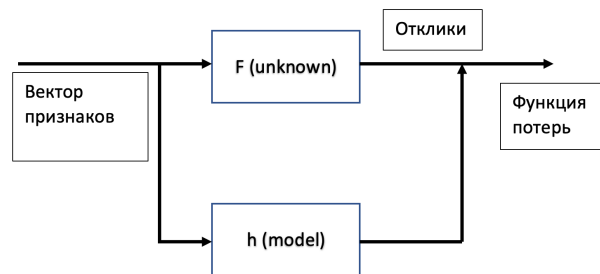


Рис. 1: Схема алгоритма обучения с учителем

При самом процессе обучения модель может быть недостаточно обучена или переобучена. Недообучение возникает в том случае, когда модель не способна точно предсказывать отклики и моделировать закономерности в данных. Когда модель настроена на избыточную классификацию, тогда можно сказать, что модель переобучена.

Проверка модели на недообучение и переобучение происходит на тестовых данных. Строится распределение плотности вероятности для тестовых и обучающих данных (т.е. распределение откликов), и далее эти распределения сравниваются.

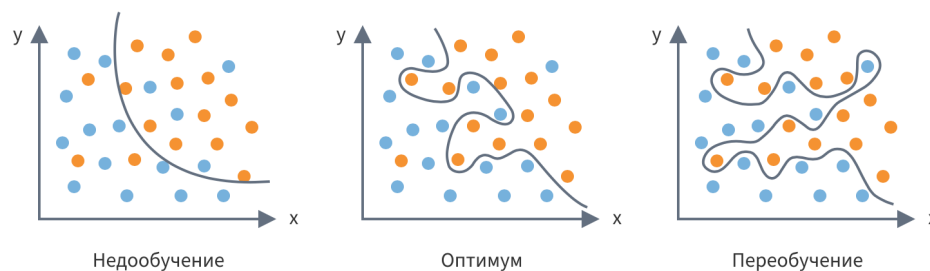


Рис. 2: Примеры недообучение и переобучение

Совокупность объектов представляет собой матрицу, которую обычно называют матрицей объектов-признаков. Если две строки матрицы линейно зависимы, то ранг матрицы уменьшается на 1. Это означает отсутствие единственности решения. Часто признаки объектов могут быть мультиколлинеарны (приблизительно линейно зависимы). На это обычно указывает неадекватность знаков или величин весов. Для того, чтобы справиться с проблемой мультиколлинеарности, задачу регуляризируют. То есть добавляют в функцию потерь слагаемое, который ставит дополнительные ограничения на вектор весов.

1.1 Бинарная классификация

Задача идентификации частицы – это задача бинарной классификации. То есть отклики в данной модели дискретны, они могут принимать только два значения, в нашем случае – сигнал или фон. Задача бинарной классификации [1] состоит в нахождении гиперплоскости, размерность которой равна $N - 1$, где N - количество признаков. Эта гиперплоскость должна оптимально разделять объекты одного класса от другого, то есть отличать сигнал от фона.

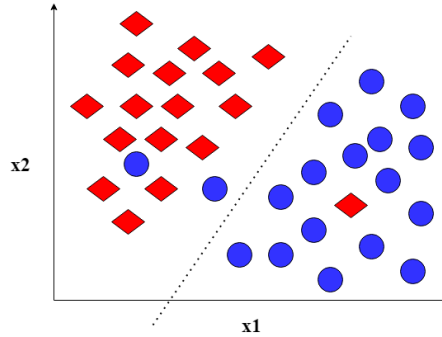


Рис. 3: Бинарная классификация

Функция потерь в данной задаче – это некоторая мера неточности классификатора в предсказании метки класса для объекта. Меньшие потери означают большую точность в предсказании метки класса. В задаче бинарной классификации функция потерь определяется как функция от разности между целями и выходами модели. То есть если объект классифицирован верно – вклад его в функцию потерь будет минимален, если классификатор ошибся, то вклад в потери будет большим. Однако среднего значения функции потерь по выборке (эмпирического риска) не достаточно для оценки ошибки модели из-за дискретного характера откликов. Поэтому необходимы дополнительные метрики для оценивания качества классификатора.

Одними из таких характеристик качества классификации являются ROC-кривая и AUC. ROC-кривая – это график зависимости доли истинных положительно определенных результатов от доли ложных положительно определенных результатов для различных значений контрольных точек проверки гипотезы. AUC – это площадь под ROC-кривой, которая является статистической характеристикой того, насколько доля истинных положительно определенных результатов превышает ложные положительно определенные результаты. То есть чем лучше классификатор разделяет два класса, тем больше AUC и тем выше ROC-кривая.

Для решения задачи бинарной классификации существует множество алгоритмов: от простых моделей, как логистическая регрессия, до глубоких нейронных сетей. В работе использованы алгоритмы BDT, MLP и DNN, речь о которых пойдет далее.

1.2 BDT

Один из подходов к решению задачи бинарной классификации – это построение деревьев решений. Идея этого алгоритма заключается в разбиении пространства признаков на множество непересекающихся секторов и дальнейшей постановке этим секторам значений моделируемой переменной. Для такого разбиения необходимы некоторые правила, следуя которым объект попадает в некоторую область. Совокупность этих правил как раз можно представить в виде дерева, во внутренних нодах (вершинах решений) которого будут и стоять эти правила. Такие деревья называются деревьями решений (decision tree). Каждый путь от корня дерева к его листу соответствует разделяемому пространству признаков сектору. Прогнозная метка класса – это метка класса с наибольшей долей примеров в пределах области, соответствующей листу, т.е. конечной вершине.

Для построения дерева решений обычно используется жадный алгоритм, заключающийся в следующем:

1. Начальное дерево содержит корень
2. Потом этот корень разветвляется на два листа, если не выполнен критерий останова

- Далее каждый нод разбивается на два дочерних листа до тех пор, пока не выполнен критерий останова

Правило ветвления должно определяться так, чтобы среди различных ветвлений выбрать то, которое максимально сокращает неоднородность в областях пространства признаков. Показателями неоднородности в случае бинарной классификации обычно выбирают индекс Джини. Он определяется следующим образом:

$$G(R_l) = \sum_{k=1}^K p_{lk}(1 - p_{lk}), \quad (1)$$

где p_{lk} - это вероятность k -го класса в области R_l , K - это количество меток класса. То есть, когда индекс Джини равен 0, можно сделать вывод, что данная область содержит объекты только одного класса.

Критерий останова обычно определяют по разному, в зависимости от необходимого результата. Это могут быть как максимальная глубина дерева, так и максимальное количество листьев или минимальное значение среднего от функции потерь в листьях.

Для повышения точности решения используют ансамбли деревьев решений. Подход, который будет рассмотрен далее, это бустинг.

Идея бустинга заключается в следующем: слабые модели в ансамбль добавляют последовательно и так, чтобы каждая модель корректировала ошибки предыдущей модели.

В работе будут рассмотрены два подхода: адаптивный бустинг и градиентный бустинг. Эти методы отличаются тем, как именно добавляемая слабая модель корректирует ошибки предыдущей модели.

Принципы адаптивного бустинга [2] заключаются в следующем:

- Каждое неглубокое дерево обучается на случайной подвыборке из обучающих данных;
- Наблюдениям приписываются веса в зависимости от точности их моделирования, т.е. наблюдениям с большей ошибкой приписывают больший вес. Этот вес наблюдения определяет вероятность попадания этого события в обучающую выборку для последующего дерева;
- Слабым моделям в ансамбле приписывается вес в соответствии со степенью доверия к ним;
- Выход ансамбля деревьев вычисляется как усреднение выходов слабых моделей.

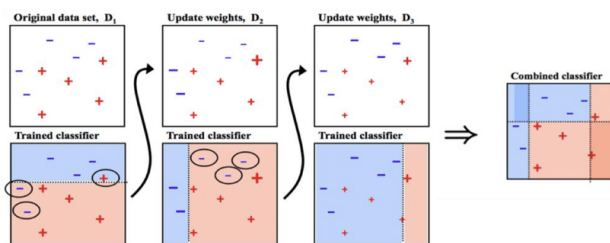


Рис. 4: AdaBoost

В отличие от адаптивного бустинга, в градиентном бустинге веса для обучающей выборки не обновляются, а вместо этого каждая слабая модель обучается с использованием остаточных ошибок предыдущей модели (псевдоостатках). Псевдоостатки - это отрицательный градиент функции потерь по выходу модели.

$$g_i = -\frac{\partial L(y_i, \bar{y}_i)}{\partial \bar{y}_i}, \quad (2)$$

где L - функция потерь, y_i - цель из обучающей выборки, \bar{y}_i - выход модели.

Базовый алгоритм градиентного бустинга заключается в следующем.

- Ансамбль инициализируется константой;
- Вычисляются псевдоостатки g_i ;
- Далее слабая модель h_j обучается на выборке вида (признаки, псевдоостатки), т.е. модель обучается предсказывать псевдоостатки ;

- После этого находится мультипликатор так, чтобы уменьшить потери на обучающей выборке. Мультипликатор задается формулой:

$$\gamma_i = \operatorname{argmin}_{\gamma} \sum_{i=1}^n L(y_i, \bar{y}_i + \gamma h_j(x_i)) \quad (3)$$

- Модель, умноженная на мультипликатор, добавляется в ансамбль;
- И так повторяется до выполнения критерия останова.

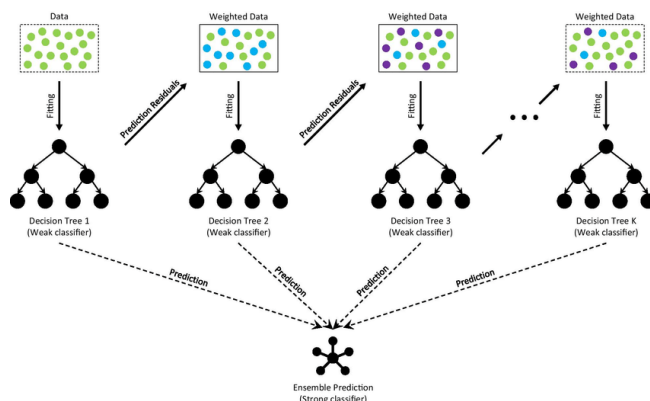


Рис. 5: Градиентный бустинг

1.3 Нейронные сети

Нейронная сеть – это сложная дифференцируемая функция, задающая отображение из признакового пространства в пространство ответов, все параметры которой могут настраиваться одновременно и взаимосвязано. Сложную функцию обычно представляют в виде композиции простых функций, которые называют слоями. Общая длина цепочки слоев определяет глубину модели. Последний слой нейронной сети называется выходным. Поскольку обучающие данные не определяют выходов каждого из слоев, кроме выходного, они называются скрытыми слоями.

Простейшие разновидности слоев:

- Линейный слой – линейное преобразование над входящими данными (его обучаемые параметры – это матрица W и вектор b). Такой слой преобразует d -мерные векторы в k -мерные.
- Функция активации [3] – нелинейное преобразование, поэлементно применяющееся к пришедшим на вход данным. Благодаря функциям активации нейронные сети способны порождать более информативные признаковые описания, преобразуя данные нелинейным образом.

Главные требования для функции активации: быть монотонной и иметь первую производную почти всюду (необходимо для обратного распространения ошибки). В качестве функции активации могут использоваться разные функции, у каждой из которых есть свои плюсы и минусы.

Сигмоида

Функция сигмоиды преобразовывает поступающие в неё значения в вещественный диапазон $[0, 1]$. То есть, если входные данные окажутся большими положительными значениями, то после преобразования они будут равны примерно единице, а отрицательные числа станут близки к нулю. Это довольно популярная функция, которую можно интерпретировать как частоту возбуждения нейрона. Однако сигмоида имеет несколько недостатков. Во-первых, область значений данной функции смещена относительно 0. Во-вторых, на хвостах сигмоиды происходит затухание градиента, что неприятно для обратного распространения ошибки. В-третьих, просчет экспоненты вычислительно сложен.

Гиперболический тангенс

Гиперболический тангенс используется в случаях, когда необходимо ограничить выходные данные в диапазоне от -1 до 1. Он часто применяется в скрытых слоях нейронных сетей. В отличие от сигмоиды

он не смещен относительно 0. Однако значения градиента при обратном распространении по-прежнему могут обнуляться. Тем не менее, использование тангенса обычно более предпочтительно.

ReLU

ReLU является наиболее популярной функцией активации и широко используется в современных нейронных сетях. Она позволяет избежать проблемы затухания градиента. Она вычисляет функцию $f(x) = \max(0, x)$, то есть просто выдаёт значения «ноль» и «не ноль».

Кроме того, ReLU очень просто вычисляется: примерно в шесть раз быстрее сигмоиды и тангенса. Однако, в ней отсутствует нулевое центрирование.

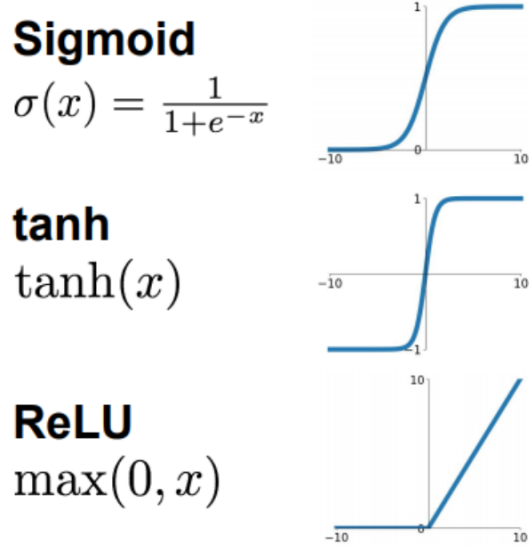


Рис. 6: Функции активации

Нейронную сеть, в которой есть только линейные слои и различные функции активации, называют многослойным персептроном (MLP).

Метод обратного распространения ошибки

Нейронная сеть обучается с помощью какой-либо модификации градиентного спуска. А для этого ей необходимо вычислять градиенты от функции потерь по всем обучающим параметрам. Нейронную сеть, как сложную функцию, можно представить в виде вычислительного графа, в которых узлы – это вычислительные операции или простые функции.

Применение нейронной сети к данным (вычисление выхода по заданному входу) называют прямым проходом. На этом этапе происходит преобразование исходного представления данных в выходное и последовательно строятся промежуточные представления данных — результаты применения слоёв к предыдущим представлениям. При обратном проходе информация движется от финального представления (от функции потерь) к исходному через все преобразования. Механизм обратного распространения ошибки, играющий важнейшую роль в обучении нейронных сетей, как раз предполагает обратное движение по вычислительному графу сети.

Метод обратного распространения ошибки заключается в рекурсивном использовании правила дифференцирования сложной функции для вычисления градиента в каждом узле, начиная с конечного узла. Информацию о величине градиента для каждого из узлов мы берем из forward pass.

2 Дискриминирующие переменные

Первые два выбранных признака [4], которые будут поданы на вход алгоритмов МО – это инвариантная масса струи и поперечный импульс струи. Ранее для выделения W-струй из большого фона КХД использовалась переменная D2. Определение переменной в [4]:

$$D_2 = E_{CF3} \times \left(\frac{E_{CF1}}{E_{CF2}} \right)^3, \quad (4)$$

где энергетические корреляционные функции задаются формулами:

$$E_{CF1} = \sum_i^n p_{T,i}; E_{CF2} = \sum_{i,j}^n p_{T,i} p_{T,j} \Delta R_{ij}; E_{CF3} = \sum_{i,j,k}^n p_{T,i} p_{T,j} p_{T,k} \Delta R_{ij} \Delta R_{jk} \Delta R_{ki}. \quad (5)$$

ΔR_{ij} определяется как:

$$\Delta R_{ij}^2 = (\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2. \quad (6)$$

Также иногда вместо переменной D2 для выделения толстой струи используют переменную C2, которая задана следующей формулой:

$$C_2 = \frac{E_{CF1} \times E_{CF2}}{E_{CF3}^2} \quad (7)$$

В работе использованы данные переменные:

1. Энергетические корреляционные функции $D2, C2, ECF2, ECF3$
2. Масса и поперечный импульс струи p_t, m
3. Planar flow (измеряет степень, в которой энергия струи равномерно распределяется по плоскости поперек поверхности струи по сравнению с линейной распределением по поверхности струи)
4. N-subjettiness (для эффективного подсчета подструй в струе) τ_1
5. Апланарность A
6. Момент Фокса-Вольфрама R_2
7. Angularity (переменная, чувствительная к степени симметрии потока энергии внутри струи)
8. KtDR
9. Разделяющая мера d_{12}

3 Процесс работы и результаты

3.1 Построение сигнального и фонового деревьев

Работа проводится с данными, полученными методом МК моделирования протон-протонного столкновения в детекторе ATLAS на LHC с энергией в системе центра масс 13 ТэВ.

Сигнальное дерево строится на данных фонового процесса образования пары топ-анти топ, так как топ преимущественно распадается через W -бозон. Ограничения, поставленные на сигнальное дерево, схожи с контрольной областью пары топ-анти топ: отбираются события с как минимум одной b -меченной струей, поперечный импульс которой больше 30 ГэВ и модуль псевдобыстроты меньше 2.5.

В фоновом дереве не должно быть W , распадающегося по адронной моде. Поэтому фоновое дерево строится на данных фонового процесса распада Z бозона в электрон-позитрон. Также поставлены ограничения на отсутствие b -меченных струй в событиях данного дерева.

И на фоновое, и на сигнальные деревья наложены ограничения на поперечный импульс толстой струи и на ее инвариантную массу. Ограничения представлены ниже:

$$60\text{GeV} < m < 110\text{GeV}, p_t > 200\text{GeV}$$

Для некоторых событий вычисление $ECF3$ не возможно. Переменные D2, C2 задавались по формулам, поэтому для событий с $ECF3$ равным 0 значения этих переменных были не определены. Для дальнейшей работы с данными и для последующего обучения моделей необходимо было удалить события, содержащие неопределенные значения.

3.2 Обучение и тестирование моделей

Рабочие данные были разделены на две равные части: обучающую и тестовую выборки. Обучающая и тестовые выборки содержат порядка 150 тысяч событий. Используются методы BDT с алгоритмом адаптивного бустинга, BDT с алгоритмом градиентного бустинга, MLP и DNN. Для разных наборов гиперпараметров обучены 4 модели.

Таблица 1: Гиперпараметры BDT

Гиперпараметр	1	2	3	4	5	6	7	8	9
NTrees	850	850	900	850	850	850	700	700	800
MinNodeSize	2.5%	2.5%	2.5%	1%	2.5%	3%	3%	2.5%	5%
MaxDepth	3	5	3	3	3	3	3	3	3
nCuts	20	20	20	20	40	20	20	20	20

Таблица 2: Гиперпараметры BDTG

Гиперпараметр	1	2	3	4	5	6	7	8	9
NTrees	850	850	900	850	850	850	700	600	800
MinNodeSize	2.5%	2.5%	2.5%	1%	2.5%	3%	3%	3%	5%
MaxDepth	5	5	3	3	3	3	3	3	3
nCuts	500	20	20	20	40	20	20	20	20

Таблица 3: Гиперпараметры MLP

Гиперпараметр	1	2	3	4	5	6	7	8	9
NeuronType	tanh	ReLU	tanh	tanh	tanh	tanh	tanh	tanh	sigmoid
HiddenLayers	N+6	N+5,N	N+5,N+4	N+5,N+4	N*2,N	N*2,N*2	N*3,N	N*2,N-1	N*2,N-1
NCycles	600	600	600	600	700	700	600	600	600
TestRate	5	5	5	5	5	5	5	5	10

В DNN в 9 случаях три скрытых слоя и выходной слой - линейный.

Таблица 4: Гиперпараметры DNN

Гиперпараметр	1	2	3	4	5	6	7	8	9
Layout	TANH	RELU	TANH	TANH	TANH	TANH	TANH	TANH	RELU
LearnRate	1e-2	1e-2	1e-2	1e-2	2e-2	2e-2	2e-2	2e-2	1e-2
BatchSize	100	100	100	100	200	200	400	800	800
Regularisator			L2	L2	L2	L2	L2	L2	L2
Drop.Config.	0.5*3	0.5*3	0.5	0.5*3	0.5*3	0	0	0	0

3.3 Резюме результатов моделей

Значения AUC для каждого из 9 случаев представлены ниже.

Таблица 5: Значения AUC

Метод	1	2	3	4	5	6	7	8	9
BDT	0.815	0.824	0.816	0.815	0.815	0.816	0.815	0.814	0.814
BDTG	0.847	0.834	0.822	0.824	0.823	0.821	0.820	0.819	0.811
MLP	0.815	0.815	0.816	0.816	0.816	0.816	0.816	0.816	0.816
DNN	0.794	0.783	0.790	0.777	0.791	0.805	0.808	0.810	0.812

Далее будут представлены характеристики моделей и описание результатов для более показательных случаев.

Первый случай

Построены Рос-кривые для четырех методов.

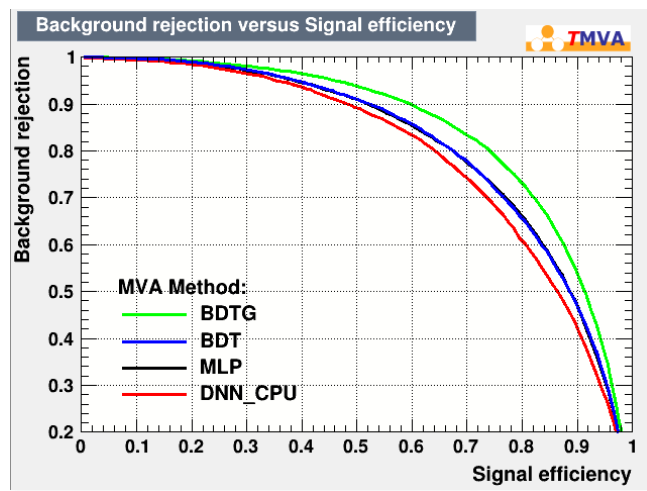


Рис. 7: ROC-кривые

По характеру полученных кривых можно сделать вывод, что классификатор BDT с градиентным бустингом лучше разделяет два класса, так как кривая этого метода лежит выше остальных.

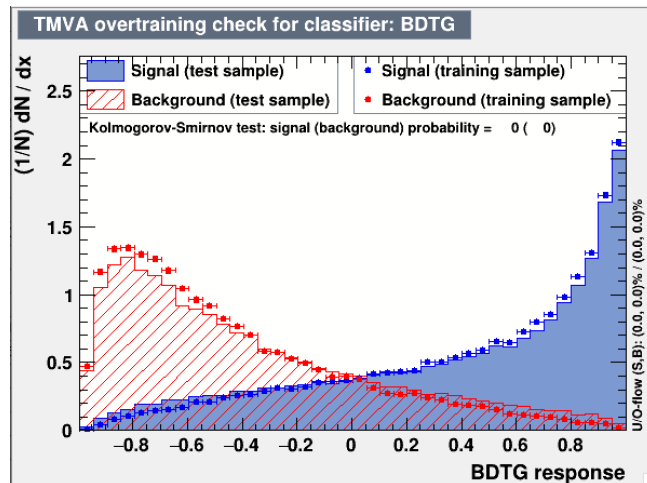


Рис. 8: Проверка на переобучение для BDTG

Однако при анализе гистограмм с распределением откликов для обучающей и тестовой выборок

можно сказать, что при данной настройке гиперпараметров модель переобучена. Распределение для обучающей выборки лежит выше распределения для тестовой выборки и не совпадает в пределах погрешности. Для остальных методов тренда на переобучение по полученным распределениям не было выявлено.

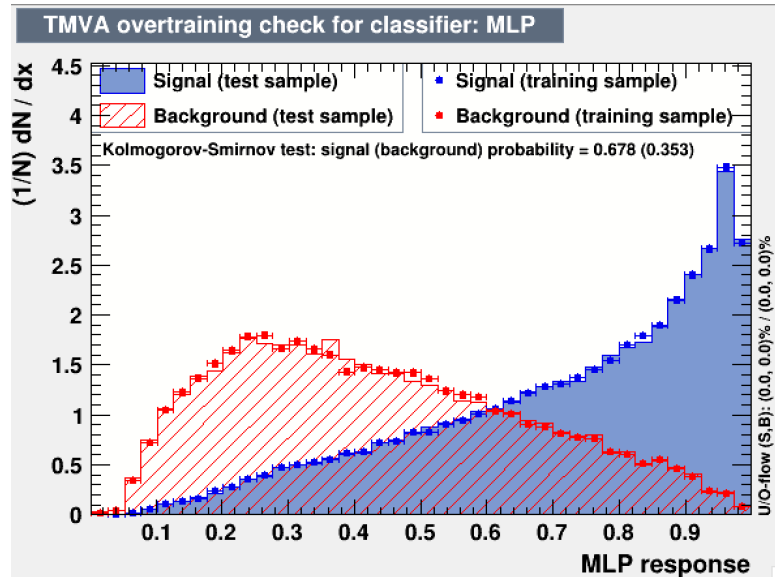


Рис. 9: Проверка на переобучение MLP

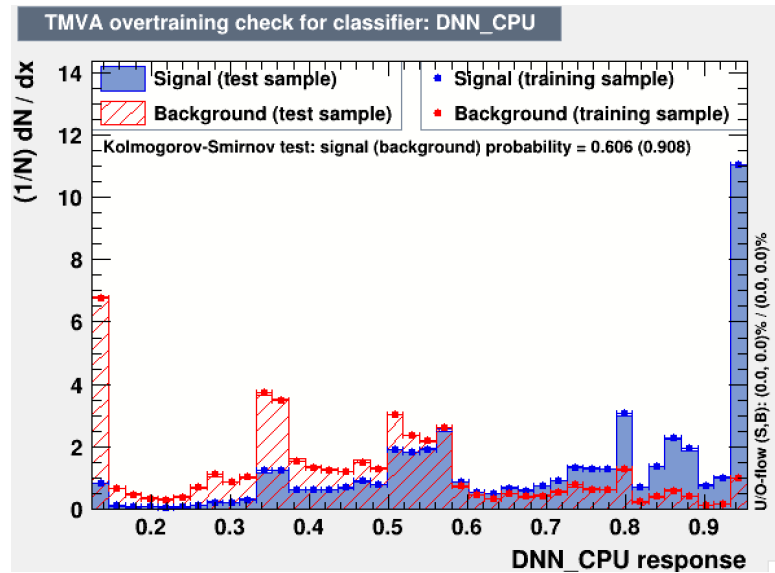


Рис. 10: Проверка на переобучение DNN

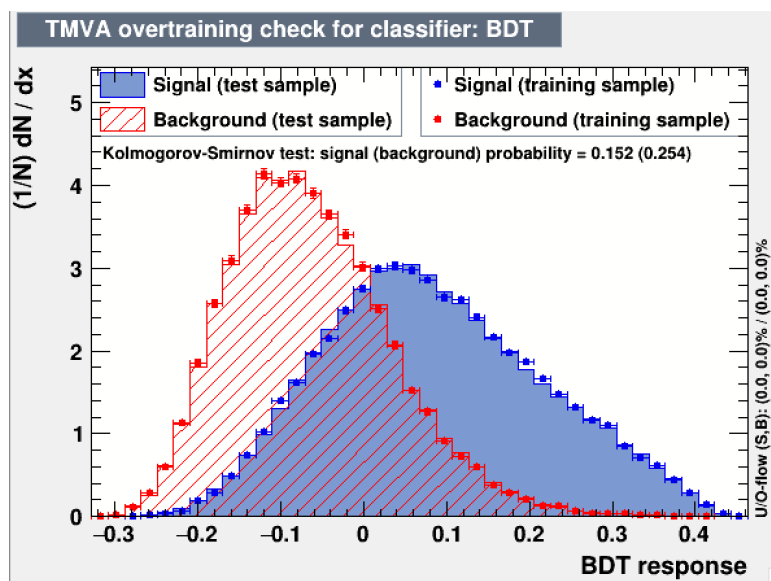


Рис. 11: Проверка на переобучение BDT

Далее после различных вариаций настроек гиперпараметров для BDTG сделан вывод о том, что модель очень чувствительна к выбору гиперпараметров, в отличие от BDT с адаптивным бустингом. Метод BDTG склонен к переобучению при неудачном выборе гиперпараметров.

Что касается сети MLP, модель кажется не сильно чувствительной к изменениям гиперпараметров. Для различного количества слоев и нейронов в слоях модель дает одинаковые результаты. Изменения функций активации также не показывает никаких ухудшений или улучшений в результатах модели.

Для предотвращения проблем с переобучением для DNN в последующих случаях решено было использовать L2-регуляризацию. Также с целью улучшения качества классификации модели решено было увеличить батч, а также убрать отсев нейронов в каждом слое. Также для каждого скрытого слоя изменена функция активации с гиперболического тангенса на ReLU.

Девятый случай Построены Roc-кривые для четырех методов.

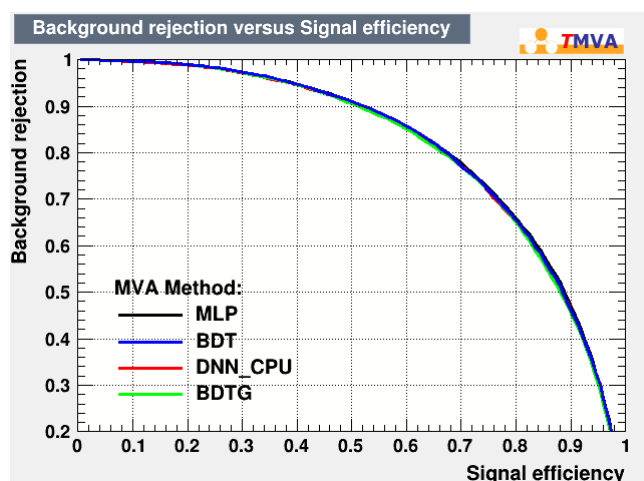


Рис. 12: ROC-кривые

По характеру полученных кривых можно сделать вывод, что результаты классификаторов одинаковые, так как кривые методов совпадают.

Также для методов тренда на переобучение по полученным распределением не было выявлено.

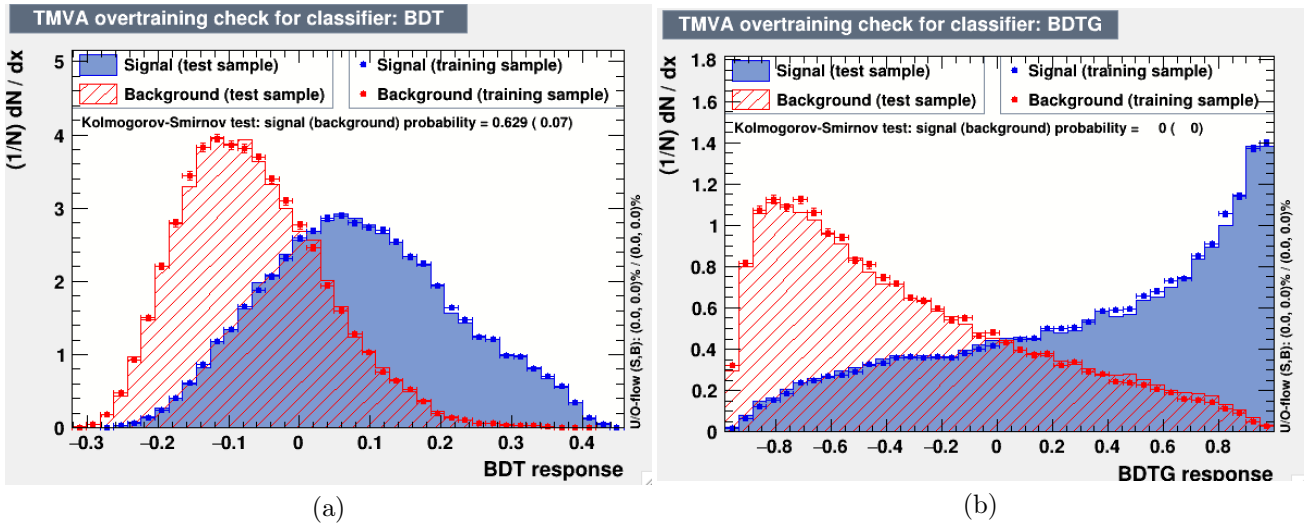


Рис. 13: Проверка на переобучение DNN(a) и MLP(b)

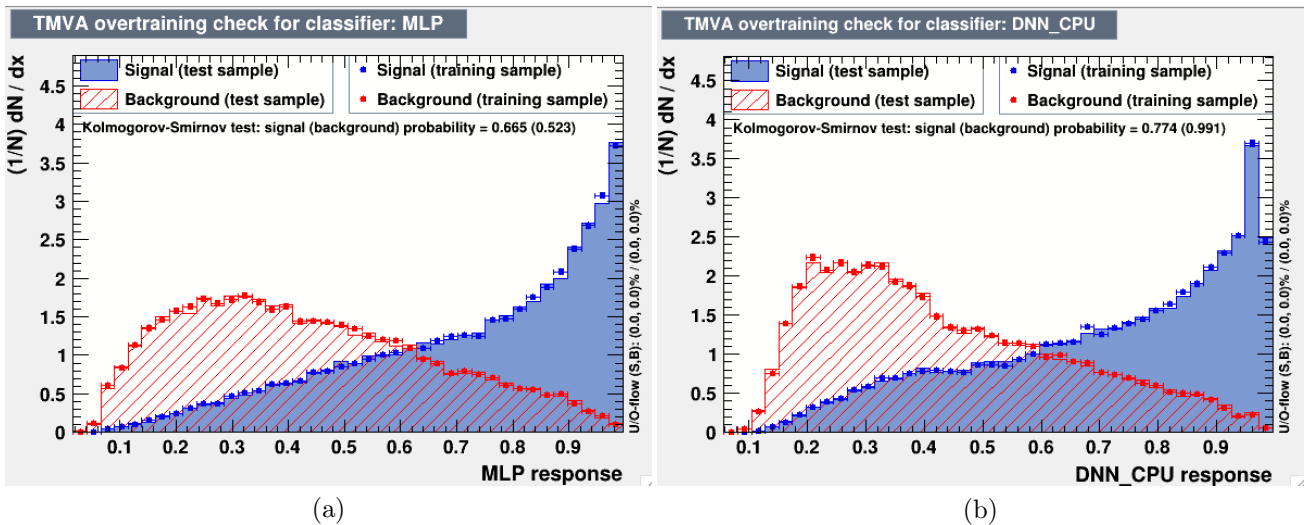


Рис. 14: Проверка на переобучение BDTG(a) и BDT(b)

4 Заключение

В рамках НИР за семестр проведено ознакомление с основными алгоритмами машинного обучения для решения задачи бинарной классификации; сформированы сигнальные и фоновые деревья, содержащие выбранные дискриминирующие переменные для данной задачи. Проведено обучение и тестирование моделей для разных наборов гиперпараметров.

На основании полученных результатов сделан вывод о том, что несмотря на интерпретируемость и быстрый темп обучения BDT с адаптивным бустингом и градиентным бустингом, дальнейшая работа будет больше сконцентрирована на обучении глубоких нейронных сетей из-за лучшей гибкости. В дальнейшем планируется расширить ассортимент алгоритмов глубокого обучения с использованием других библиотек в Python.

Список литературы

- [1] Школа Анализа Данных. Учебник по машинному обучению. <https://education.yandex.ru/handbook/ml/article/about>.
- [2] Andreas Hocker et al. TMVA - Toolkit for Multivariate Data Analysis. 3 2007.
- [3] Stanford University School of Engineering. Свёрточные нейронные сети для визуального распознавания. <https://www.reg.ru/blog/stenfordskij-kurs-lekciya-1-vvedenie/>.
- [4] Identification of Hadronically-Decaying W Bosons and Top Quarks Using High-Level Features as Input to Boosted Decision Trees and Deep Neural Networks in ATLAS at $\sqrt{s} = 13$ TeV. Technical report, CERN, Geneva, 2017. All figures including auxiliary figures are available at <https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2017-004>.