Abstract geometric lines in the top-left corner of the slide, consisting of several overlapping, irregular polygons and lines in a light gray color.

MALWARE DETECTION IN NETWORK TRAFFIC DATA

О ДАННЫХ

Каждая строка описывает сетевой поток более чем с 20 атрибутами: IP-адреса и порты, протокол, длительность, счетчики пакетов/байтов, состояние сеансов и другими

Датасет снабжен двумя уровнями разметки:

- label – Benign или Malicious(будет целевой переменной);
- detailed label – 10 конкретных подтипов атак (используется лишь для валидации и анализа ошибок, но не для обучения)

РАЗВЕДОЧНЫЙ АНАЛИЗ ДАННЫХ (EDA)

Сразу же удалим столбцы, которые нам вряд ли пригодятся:

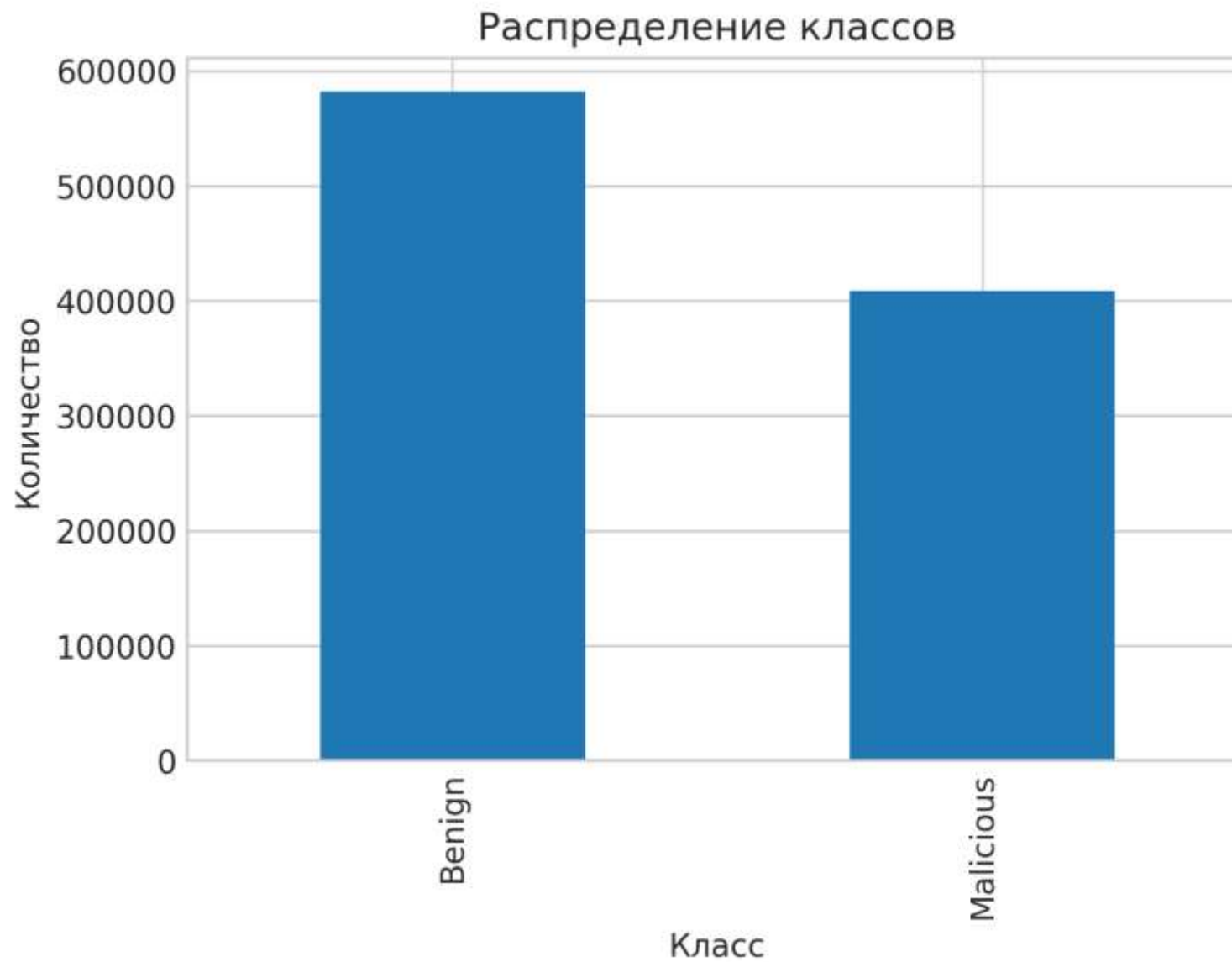
- **ts** - временная метка события подключения
- **uid** - уникальный идентификатор соединения
- **id.orig_h** - исходный IP-адрес
- **id.resp_h** - IP-адрес назначения
- **local_orig** и **local_resp** – указывает, считается ли соединение локальным или нет
- **missed_bytes** – количество пропущенных байтов в соединении
- **tunnel_parents** – указывает, является ли это соединение частью туннеля
- **detailed-label** – более подробное описание или метка соединения


КАК МЫ ПОЛУЧИЛИ ШИКАРНЫЙ ДАТАСЕТ?

Далее делаем следующие действия над датасетом:

1. Приводим в порядок типы данных в каждом столбце
2. Уменьшим количество данных в 15 раз (изначально была взята большая выборка, чтобы захватить больший объем данных и из него случайно выбрать меньшее количество строк)
3. Заменим все пропуски на 0 в числовых признаках
4. Все label типа 'Malicious PartOfAHorizontalPortScan', 'Malicious DDoS', 'Malicious Attack' и 'Malicious C&C' на 'Malicious' (Будем заниматься исключительно бинарной классификацией Benign/Malicious)

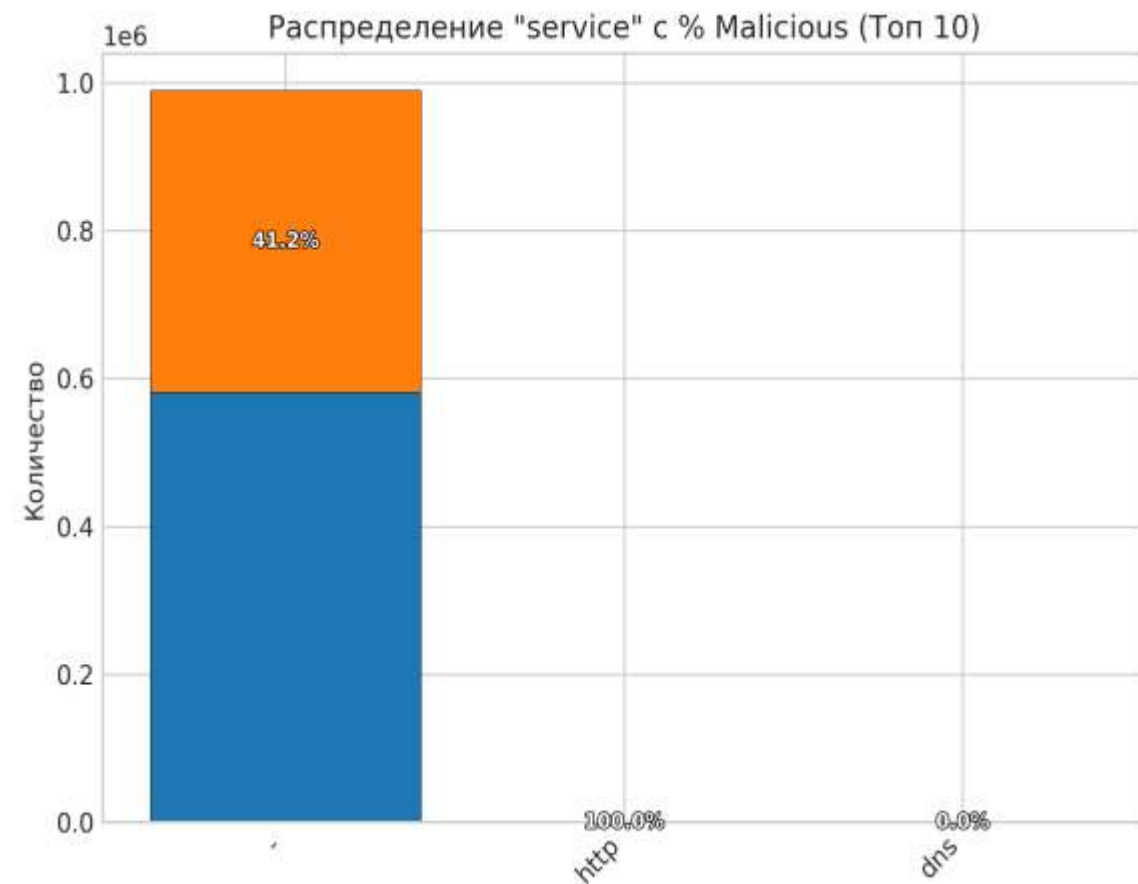
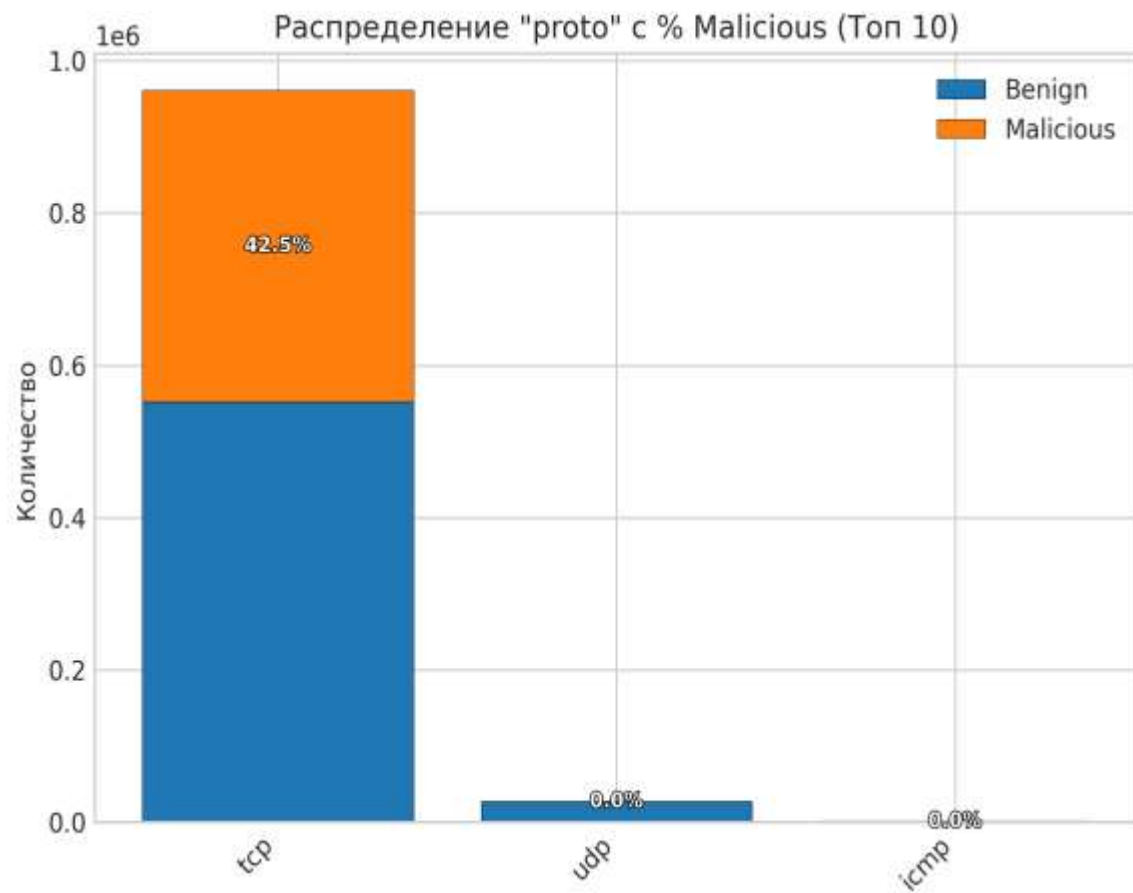
РАСПРЕДЕЛЕНИЕ ЦЕЛЕВЫХ ПРИЗНАКОВ ПО ДАТАСЕТУ





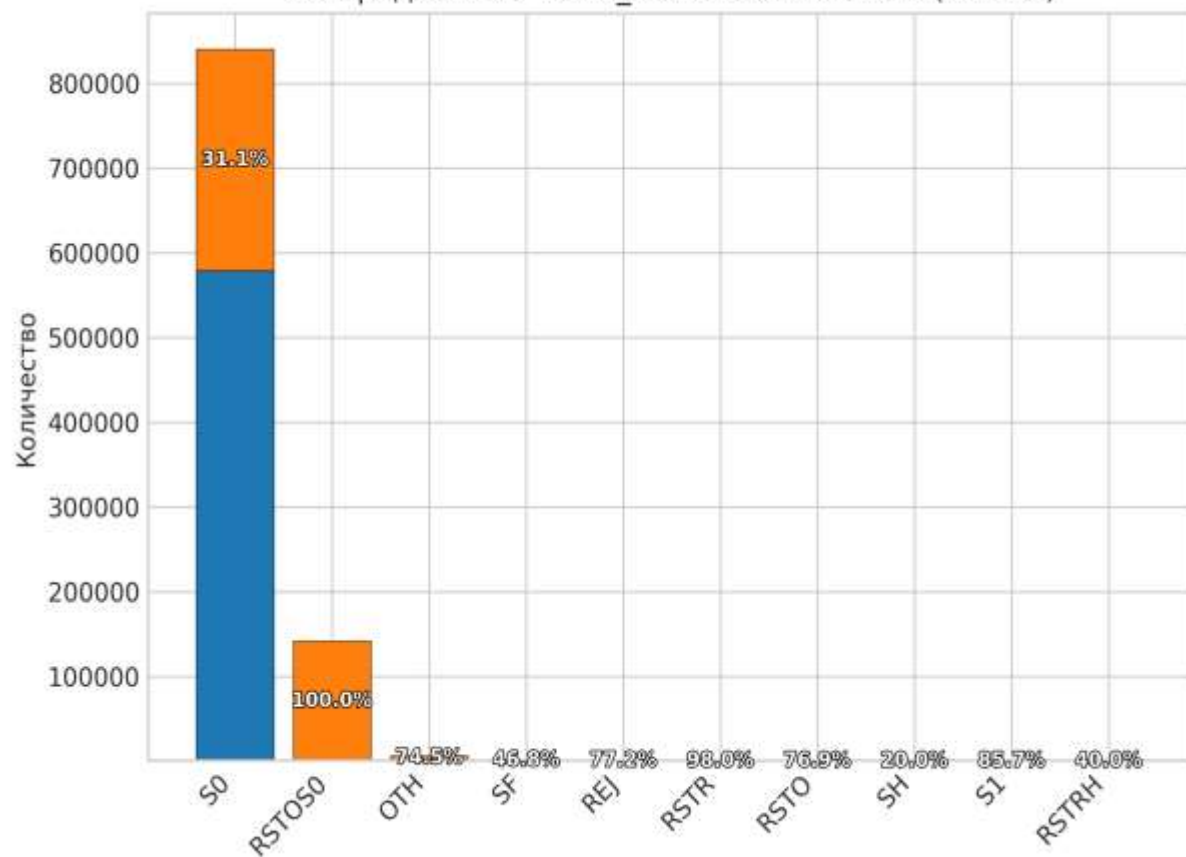
ПРО КАТЕГОРИАЛЬНЫЕ ПРИЗНАКИ

РАСПРЕДЕЛЕНИЕ 'PROTO' И 'SERVICE'

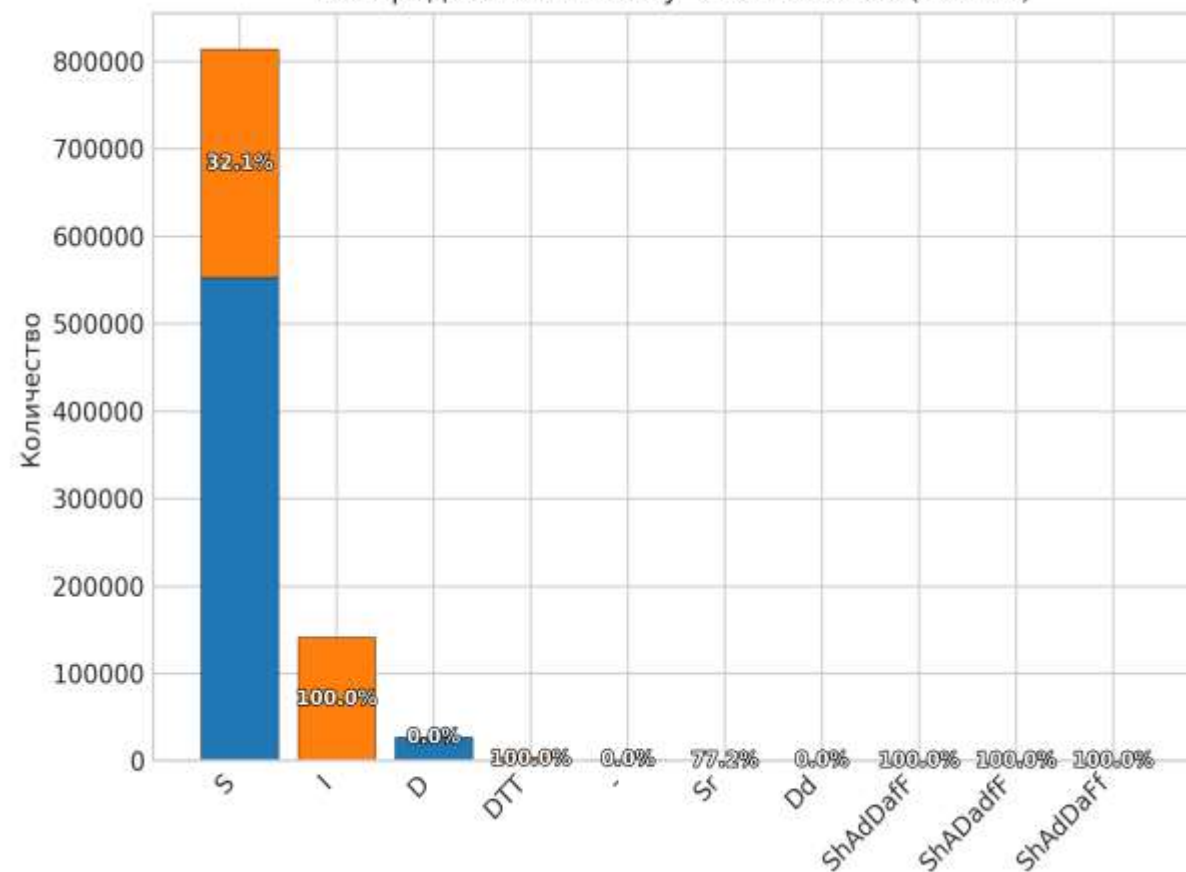


РАСПРЕДЕЛЕНИЕ 'CONN_STATE' И 'HISTORY'

Распределение "conn_state" с % Malicious (Топ 10)

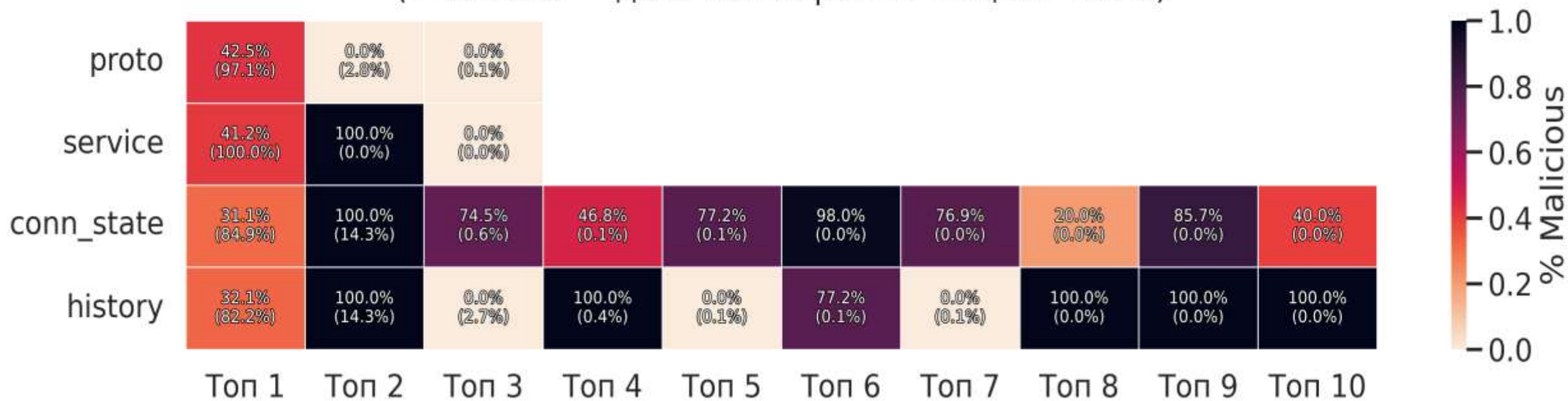


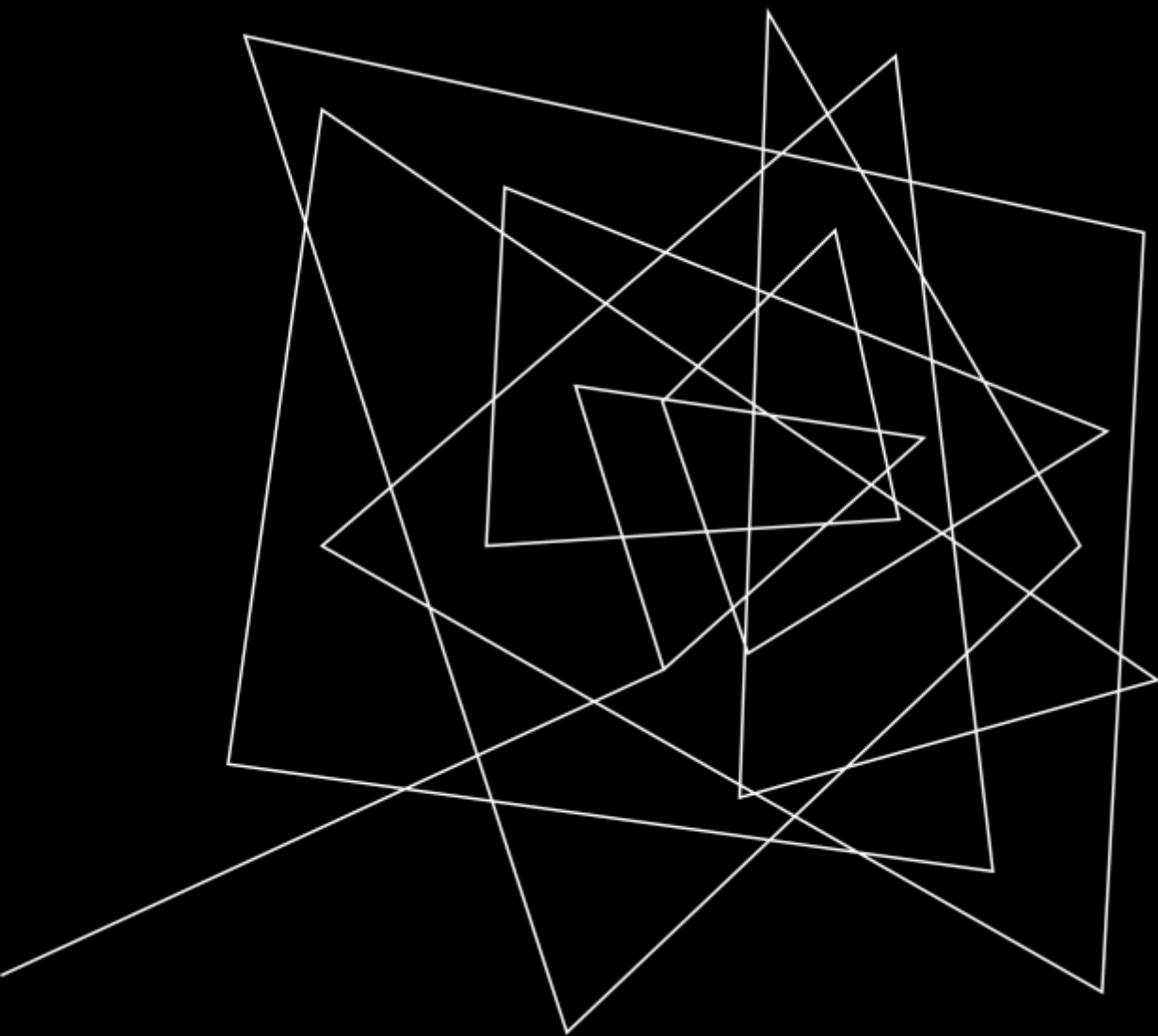
Распределение "history" с % Malicious (Топ 10)



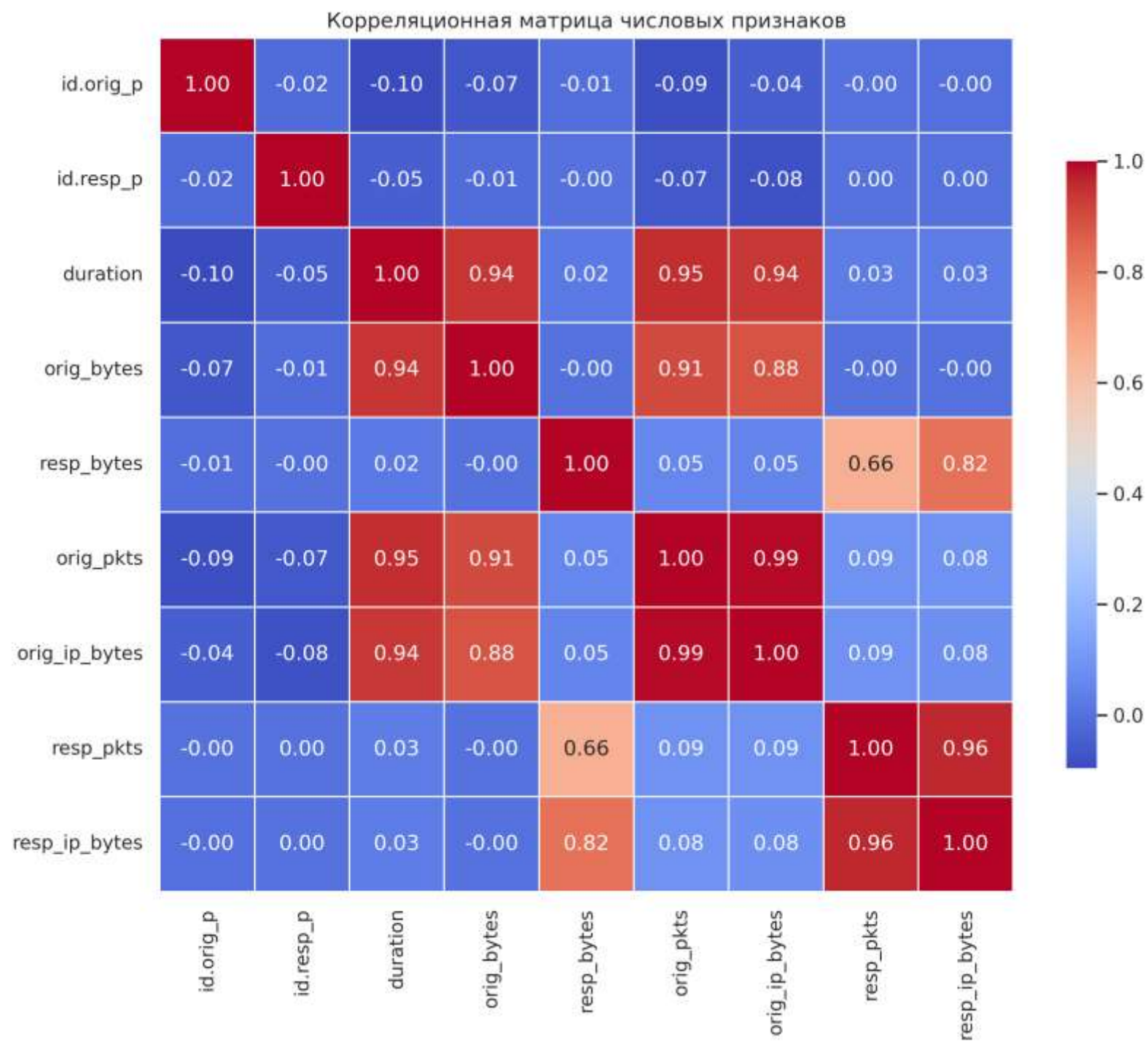
ТЕПЛОВАЯ КАРТА 'MALICIOUS'

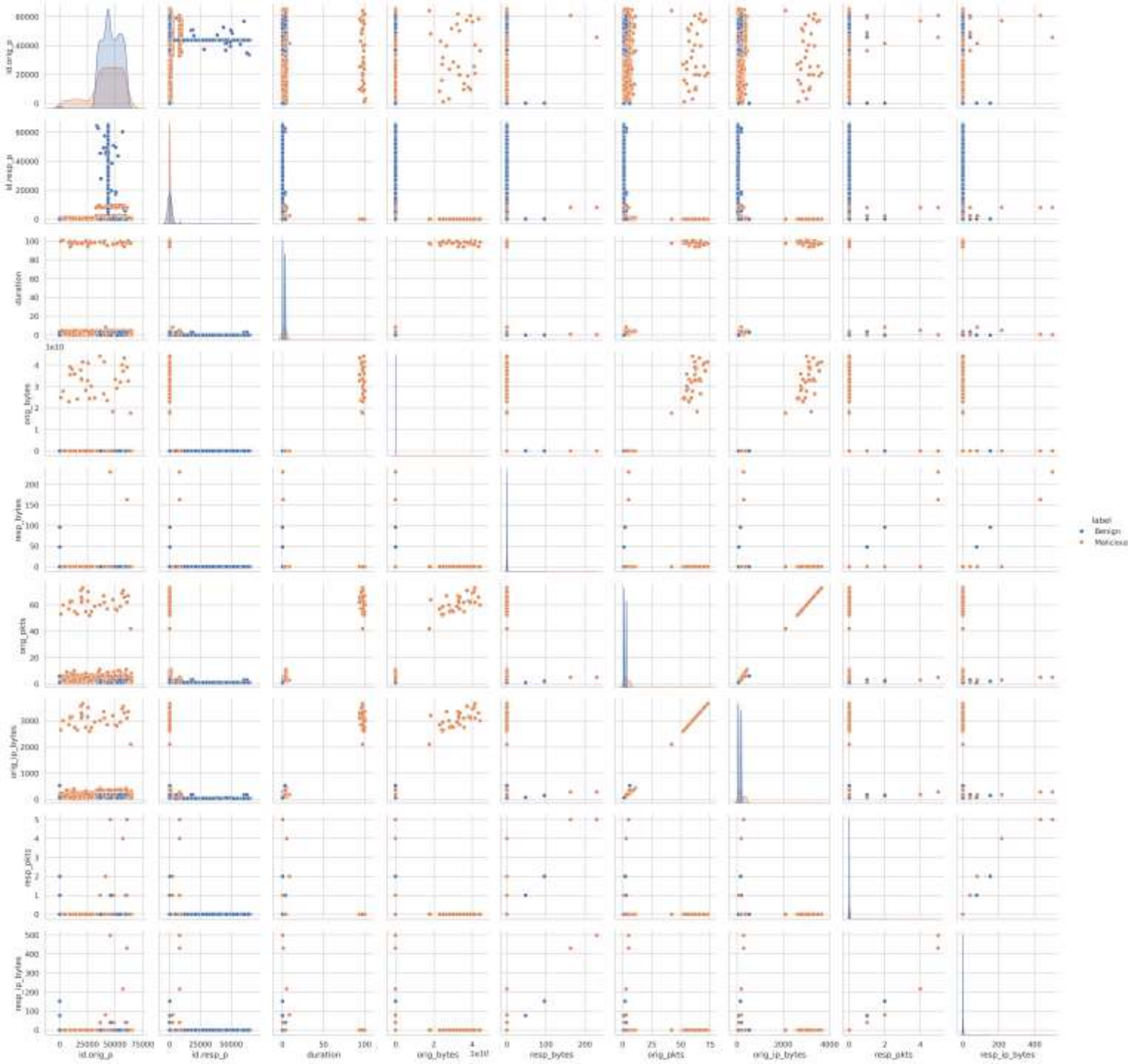
Тепловая карта Malicious
(в скобках — доля категории от общего числа)





РАСПРЕДЕЛЕНИЕ ЧИСЛОВЫХ ПРИЗНАКОВ





РАСПРЕДЕЛЕНИЕ 'BENIGN' И 'MALICIOUS' ПО ПАРАМ ПРИЗНАКОВ

ВЫВОДЫ ПОСЛЕ АНАЛИЗА ДАТАСЕТА

- Баланс классов умеренный (но с метриками все равно надо быть аккуратнее)
- По **proto** тяжело что-то сказать, а вот в **service**, **conn_state** и **history** много, практически на 100%, злонамеренных категорий, которые почти сразу можно брать за правило
- Признаки **orig_pkts**, **orig_bytes**, **orig_ip_bytes** и **duration** (так же как и **resp_pkts** и **resp_ip_bytes**) фактически дублируют друг друга (что логично, так как все эти признаки напрямую друг с другом связаны)
- Остальные числовые признаки практически не коррелируют, поэтому деревьям/градиентным бустингам будет хорошо

ПОДГОТОВКА ДАННЫХ И ОБУЧЕНИЕ МОДЕЛЕЙ

КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ

Всего 4 признака: **proto**, **service**, **conn_state**, **history**.

Для первых трёх будем использовать **OneHotEncoding**, а для **history** **OrdinalEncoding**.

Вообще говоря, так-то это и неплохо, поскольку на **history** можно ввести частичный порядок $x \leq y$ если кол-во **conn_state** в **y** больше или равно кол-ву **conn_state** в **x**

Label – тут просто: Benign=0, Malicious=1

СТАНДАРТИЗАЦИЯ ПРИЗНАКОВ

Для повышения качества моделей логистической регрессии и k-NN воспользуемся **StandardScaler**.

StandardScaler — это метод стандартизации данных, при котором данные преобразуются так, чтобы иметь нулевое среднее значение и единичное стандартное отклонение .

Формула:

где:

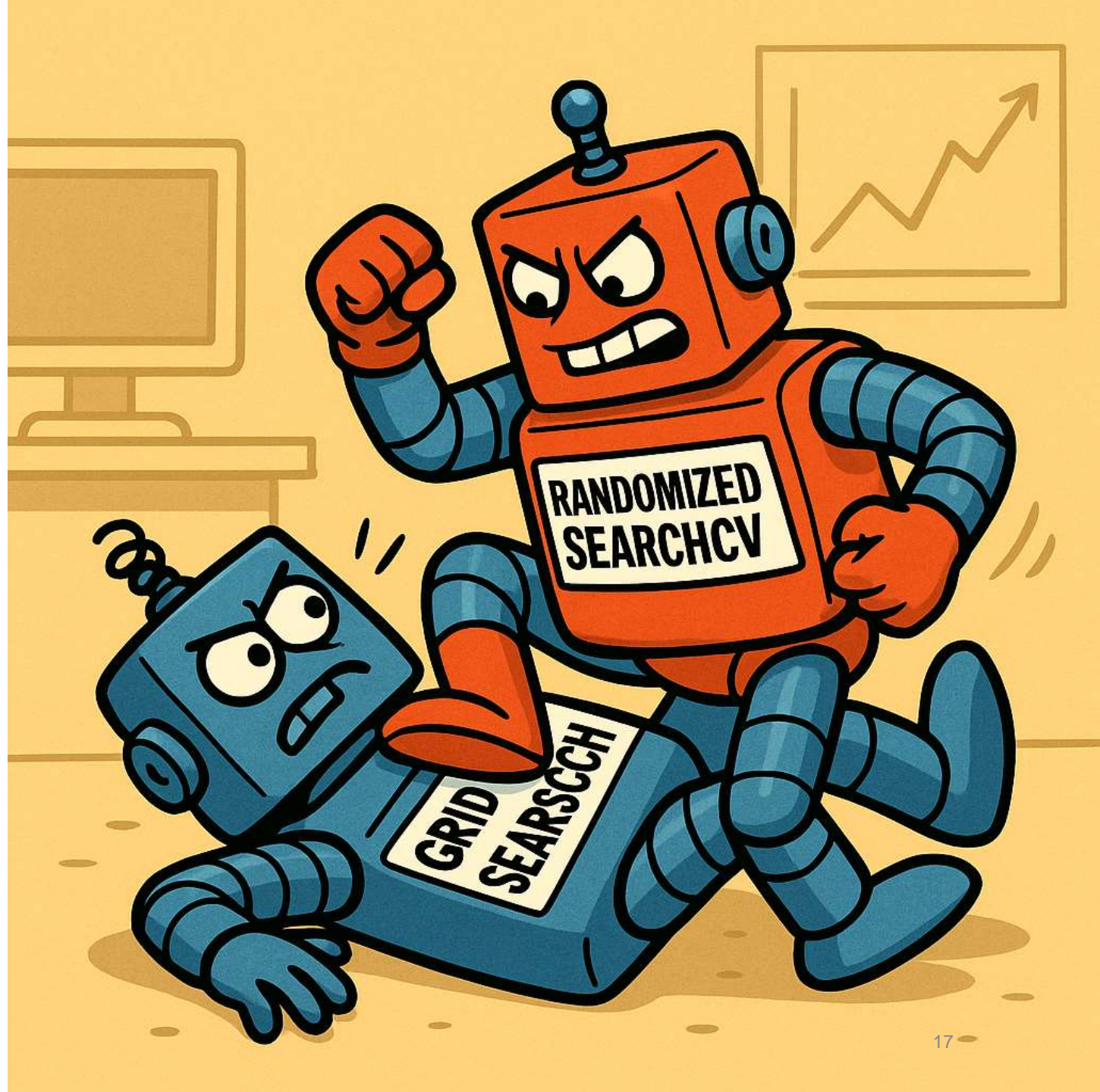
- μ — среднее значение признака,
- σ — стандартное отклонение.

$$z = \frac{x - \mu}{\sigma}$$

RandomizeSearchCV

VS

GridSearchCV



ВЛИЯНИЕ НА МЕТОДЫ:

1) Логистическая регрессия

Что происходит без масштабирования:

- Логистическая регрессия — это линейная модель , которая строит разделяющую гиперплоскость(линейная формула).

$$\sum_{j=1}^n w_j x_j + b = w^T x + b = 0$$

- Если признаки имеют разный масштаб (например, **x1** от 0 до 1, а **x2** от 0 до 1000), то коэффициенты **w1** и **w2** будут несопоставимыми по величине.
- Это затрудняет интерпретацию весов : например, большой вес у **x2** может быть просто следствием его большого масштаба, а не реальной силы влияния.
- Также это важно для корректной работы **L1/L2-регуляризации**, потому что она штрафует большие веса

2) k-NN — это метрический алгоритм , он использует расстояние между объектами (например, евклидово).

Это приведёт к тому, что модель будет игнорировать менее масштабированные признаки.

Что произойдёт без StandardScaler:

Расстояния будут искажены → выбор ближайших соседей будет некорректным → точность модели резко упадёт.



3) Когда можно не использовать **StandardScaler**?

Деревья решений , случайные леса , градиентный бустинг — эти модели не чувствительны к масштабу признаков , потому что они не используют расстояния или градиентный спуск.

Если все признаки уже в одном масштабе (например, нормализованы от 0 до 1).



ЧТО ГОВОРЯТ О НАШЕМ ДАТАСЕТЕ ВЫБРАННЫЕ ГИПЕРПАРАМЕТРЫ

1) Логистическая регрессия:

- **C=100** — сила регуляризации

Данные, скорее всего, хорошо разделимы, и модель может позволить себе "уверенные" веса без риска переобучения.

- **class_weight='balanced'** — баланс классов

Модель автоматически учитывает дисбаланс классов, увеличивая вес редких классов.

- Выводы о датасете:

Классы неравномерно распределены (например, соотношение 1:10 или хуже). Много объектов одного класса и мало другого (иначе **class_weight** не нужен).

2) KNN

- **n_neighbors=47** — большое число соседей

Алгоритм учитывает 47 ближайших объектов для классификации/регрессии.

- **weights='distance'** — веса обратно пропорциональны расстоянию
Классы неравномерно распределены — некоторые объекты могут быть "островами" среди чужих классов.

Есть выбросы — модель снижает их влияние, уделяя больше внимания ближайшим "нормальным" точкам.

- Выводы о датасете:

Данные, скорее всего, зашумлены — модель полагается на усреднение по многим точкам, чтобы снизить влияние выбросов.

Датасет достаточно большой — иначе 47 соседей было бы слишком много (например, для маленького датасета это привело бы к **underfitting**).

3) XGBoost

- **learning_rate=0.05** — медленный темп обучения

Данные, вероятно, содержат сложные зависимости, которые нельзя уловить быстро.

- **gamma=0** — минимальное сокращение потерь для разделения

Что это значит?

Деревья не штрафуются за создание дополнительных листьев.

Модель может свободно разделять узлы, даже если выигрыш минимален.

- Выводы о датасете:

Данные не требуют жёсткой регуляризации через **gamma**.

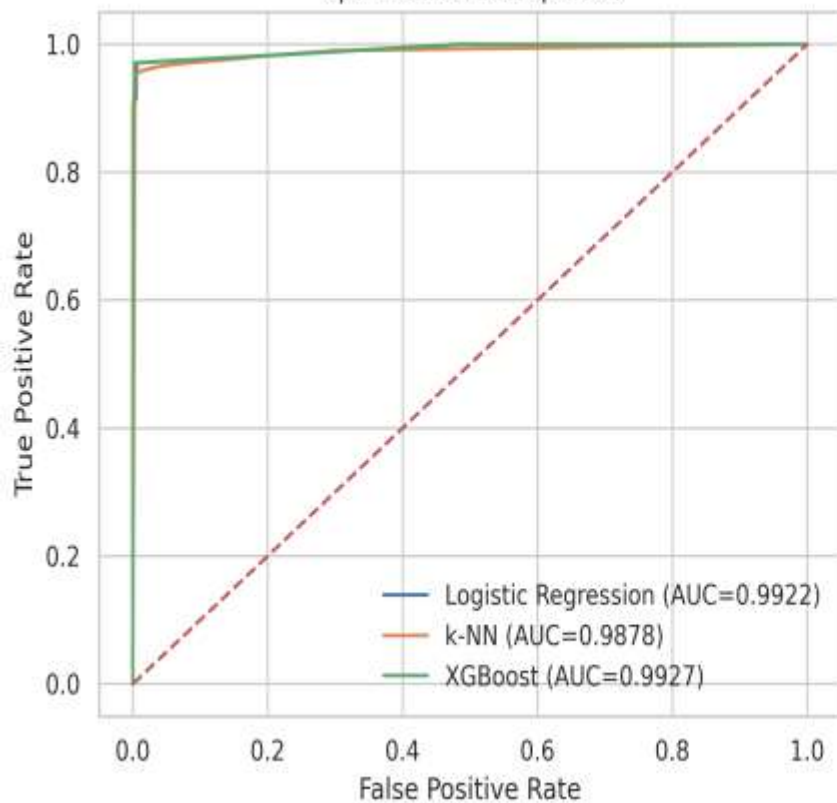
Возможно, классы хорошо разделяются без чрезмерного усложнения деревьев.

An abstract geometric design featuring two thin, dark lines that intersect on a light gray background. One line runs diagonally from the top-left towards the bottom-right, while the other runs from the top-right towards the bottom-left. The intersection point is located to the left of the text.

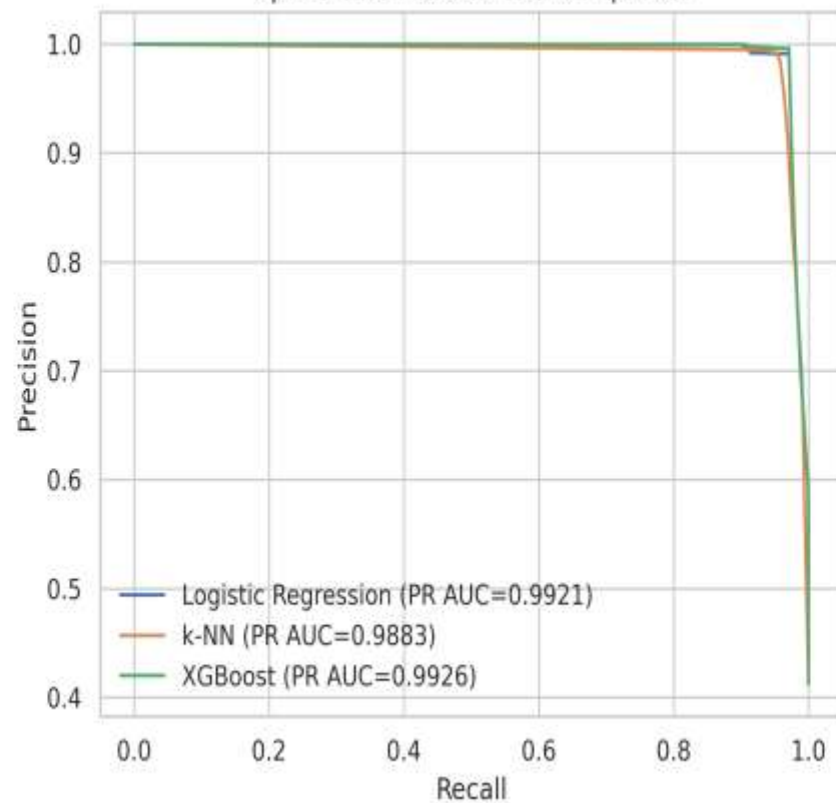
РЕЗУЛЬТАТЫ РАБОТЫ АЛГОРИТМОВ

СРАВНЕНИЕ ВИЗУАЛИЗАЦИЙ ИТОГОВ РАБОТЫ АЛГОРИТМОВ

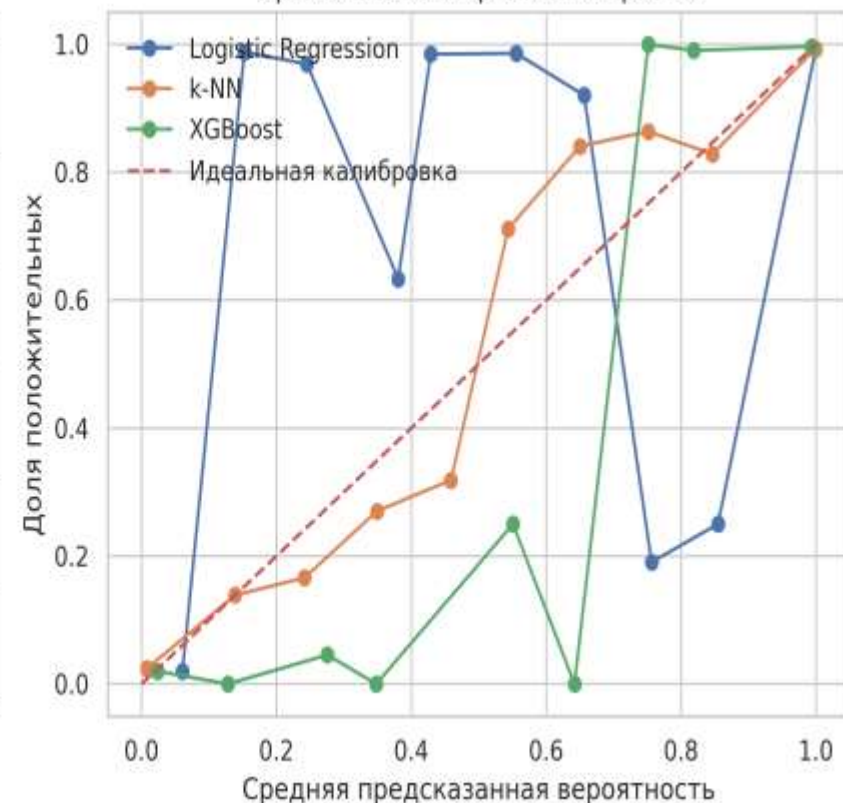
Сравнение ROC-кривых



Сравнение Precision-Recall кривых



Сравнение калибровочных кривых





ИТОГИ СРАВНЕНИЯ КАЧЕСТВА

XGBoost чуть лучше всех разделяет "хорошие" и "плохие" образцы (самая высокая ROC-AUC и минимальное число пропусков вредоносных подключений) при очень малом числе ложных тревог, тогда как логистическая регрессия почти не уступает ему по AUC, но чуть больше пропускает "плохих", а k-NN отстаёт и при этом медленнее. Поэтому наилучшим вариантом будет XGBoost.



ВЫВОД ОБ ОЦЕНКЕ ВЕРОЯТНОСТЕЙ

При этом все три модели без дополнительной калибровки переоценивают или недооценивают вероятности (лучше всех в этом плане оказался k-NN), так что для надёжного подбора порога стоит выполнить `CalibratedClassifierCV` и выбрать оптимальный:

- Жёсткий (скажем 0.2) для немедленного блокирования точно вредоносных.
- Мягкий (скажем 0.7) для ручной проверки сомнительных подключений.

НЕМНОГО О НАС:

Шевцов Владислав



Иващенко Дмитрий



Карабаев Никита



ВОТ И ВСЕ!!!!!!
СПАСИБО, ЧТО ПОСЛУШАЛИ!!!!

