Датасет NSL KDD для систем обнаружения атак является.

NSL-KDD - это набор данных, предложенный для решения некоторых неотъемлемых проблем набора данных KDD'99

Ссылка на github исходного NSL KDD: https://github.com/jmnwong/NSL-KDD-Dataset

Однако в данном репозиторие представлены файлы формата txt и arff

Ссылка на другой github в котором есть дополнительные файлы для тестов и файлы с расширением csv более удобным для использования: https://github.com/defcom17/NSL\_KDD

|  |  |  |  |
| --- | --- | --- | --- |
| Номер признака | Имя признака | Описание | Тип признака |
| ОСНОВНЫЕ ХАРАКТЕРИСТИКИ КАЖДОГО ВЕКТОРА СЕТЕВОГО ПОДКЛЮЧЕНИЯ | | | |
| 0 | Duration | Продолжительность  времени соединения | int64 |
| 1 | protocol\_type | Протокол, используемый  в соединении | symbolic |
| 2 | Service | Используемая сетевая служба | symbolic |
| 3 | flag | Состояние соединения -нормально или ошибка | symbolic |
| 4 | src\_bytes | Количество байтов данных, передаваемых от источника к месту назначения в одном соединении | int64 |
| 5 | dst\_bytes | Количество байтов данных, переданных  от места назначения к  источнику в одном соединении | int64 |
| 6 | Land | Eсли IP-адреса источника и назначения и номера портов равны, то эта переменная принимает значение 1 иначе 0 | int64 |
| 7 | wrong\_fragment | Общее количество  Неправильных фрагментов в этой связи | int64 |
| 8 | urgent | Количество срочных пакетов в этом соединении. Срочные  пакеты-это пакеты с  активированным битом срочности | int64 |
| ХАРАКТЕРИСТИКИ, СВЯЗАННЫЕ С СОДЕРЖАНИЕМ КАЖДОГО ВЕКТОРА СЕТЕВОГО ПОДКЛЮЧЕНИЯ | | | |
| 9 | hot | Количество "горячих" индикаторов в содержании, таких как:  вход в системный каталог, создание  программ и выполнение  программ | int64 |
| 10 | num\_failed\_logins | Количество неудачных попыток входа в систему | int64 |
| 11 | logged\_in | Статус входа в систему:  1 при успешном входе в систему; 0 в противном случае | int64 |
| 12 | num\_compromised | Количество  `скомпрометированных"  условий | int64 |
| 13 | root\_shell | 1, если получена корневая оболочка; 0 в  противном случае | int64 |
| 14 | su\_attempted | 1 если команда `su root"  была предпринята или  использована; 0 в противном случае | int64 |
| 15 | num\_root | Количество обращений  `root" или количество  операций, выполняемых как root в соединении | int64 |
| 16 | num\_file\_creations | Количество  операций создания файлов в соединении | int64 |
| 17 | num\_shells | Количество подсказок  оболочки | int64 |
| 18 | num\_access\_files | Количество операций с  файлами контроля доступа | int64 |
| 19 | num\_outbound\_cmds | Количество исходящих  команд в сеансе ftp | int64 |
| 20 | is\_host\_login | 1 если логин  принадлежит к  `горячему" списку, то есть root или admin;  иначе 0 | int64 |
| 21 | is\_guest\_login | 1, если логин является  `гостевым"  логином; 0 в  противном случае | int64 |
| ВРЕМЕННЫЕ ХАРАКТЕРИСТИКИ ТРАФИКА КАЖДОГО ВЕКТОРА СЕТЕВОГО ПОДКЛЮЧЕНИЯ | | | |
| 22 | count | Количество  подключений к тому  же хосту назначения, что и текущее соединение за последние две секунды | int64 |
| 23 | srv\_count | Количество  подключений к той же  службе (номер порта), что и текущее соединение за последние две секунды | int64 |
| 24 | serror\_rate | Процент соединений,  активировавших  flag (3) s0, s1, s2 или s3,  среди соединений,  объединенных в  count (22) | float64 |
| 25 | srv\_serror\_rate | Процент соединений,  активировавших  flag (3) s0, s1, s2 или s3,  среди соединений,  объединенных в  srv\_count (23) | float64 |
| 26 | rerror\_rate | Процент соединений,  активировавших  flag (3) REJ, среди  соединений, объединенных в  count (22) | float64 |
| 27 | srv\_rerror\_rate | Процент соединений,  активировавших  flag (3) REJ, среди  соединений,  объединенных в  srv\_count (23) | float64 |
| 28 | same\_srv\_rate | Процент соединений  ,  которые были подключены к одной  и той же  службе, среди  соединений,  объединенных в  count (23) | float64 |
| 29 | diff\_srv\_rate | Процент соединений,  которые были подключены к  различным службам,  среди соединений,  объединенных в  count (22) | float64 |
| 30 | srv\_diff\_host\_rate | Процент соединений,  которые были подключены к разным  целевым машинам,  среди соединений,  объединенных в  srv\_count (23) | float64 |
| ОСОБЕННОСТИ ТРАФИКА НА ОСНОВЕ ХОСТА В ВЕКТОРЕ СЕТЕВОГО ПОДКЛЮЧЕНИЯ | | | |
| 31 | dst\_host\_count | Количество  соединений,  имеющих один и тот же IP-адрес хоста назначения | int64 |
| 32 | dst\_host\_srv\_count | Количество соединений  с одинаковым  номером порта |  |
| 33 | dst\_host\_same\_srv\_rate | Процент соединений,  которые были подключены к одной и той же службе, среди  соединений,  объединенных в  dst\_host\_count(31) | float64 |
| 34 | dst\_host\_diff\_srv\_rate | Процент соединений,  которые были подключены к  различным службам,  среди соединений,  объединенных в  dst\_host\_count(31) | float64 |
| 35 | dst\_host\_same\_src\_port\_rate | Процент соединений,  которые были подключены к одному и тому же исходному  порту, среди соединений,  объединенных в  dst\_host\_srv\_count (32) | float64 |
| 36 | dst\_host\_srv\_diff\_host\_rate | Процент соединений,  которые были подключены к разным  целевым машинам,  среди соединений,  объединенных в  dst\_host\_srv\_count (32) | float64 |
| 37 | dst\_host\_serror\_rate | Процент соединений,  активировавших  flag (3) s0, s1, s2 или s3,  среди соединений,  объединенных в  dst\_host\_count(31) | float64 |
| 38 | dst\_host\_srv\_serror\_rate | Процент соединений,  активировавших  flag (3) s0, s1, s2 или s3,  среди соединений,  объединенных в dst\_host\_srv\_count (32) | float64 |
| 39 | dst\_host\_rerror\_rate | Процент соединений,  активировавших  flag (3) REJ, среди  соединений,  объединенных в dst\_host\_count(31) | float64 |
| 40 | dst\_host\_srv\_rerror\_rate | Процент соединений,  активировавших  flag (3) REJ, среди  соединений,  объединенных в dst\_host\_srv\_ount (31) | float64 |
| Целевая переменная | | | |
| 41 | type\_attack | Тип атаки | symbolic |

Столбец type\_attack содержит данные о различных 5 классах векторов сетевых подключений, и они классифицируются как один нормальный класс и четыре класса атак.

4 класса атак сгруппированы как DoS, Probe, R2L и U2R.

|  |  |
| --- | --- |
| Тип атаки | Атаки |
| DOS | Back,Land,Neptune,Pod,Smurf,Teardrop,Mailbomb,Processtable,  Udpstorm, Apache2,Worm |
| Probe | Satan,IPsweep,Nmap,Portsweep,Mscan,Saint |
| R2L | Guess\_password,Ftp\_write,Imap,Phf,Multihop,Warezmaster,Xlock,  Xsnoop,Snmpguess,Snmpgetattack,Httptunnel,Sendmail, Named,Spy, warezclient |
| U2R | Buffer\_overflow,Loadmodule,Rootkit,Perl,Sqlattack,Xterm,Ps |

1. DOS: Отказ в обслуживании – это категория атак, которая истощает ресурсы жертвы, тем самым делая ее неспособной обрабатывать законные запросы- например, syn flooding. Соответствующие функции: “source bytes ” и “ percentage of packets with errors “

2. Probing: Целью слежки и других зондирующих атак является получение информации об удаленной жертве,например, сканирование портов. Соответствующие функции: “duration of connection” and “source bytes”

3. U2R: несанкционированный доступ к привилегиям локального суперпользователя (root) - это тип атаки, при котором злоумышленник использует обычную учетную запись для входа в систему жертвы и пытается получить привилегии root/администратора, используя некоторую уязвимость в жертве, например атаки переполнения буфера. Соответствующие функции: “number of file creations” and “number of shell prompts invoked”

4. R2L: несанкционированный доступ с удаленной машины, злоумышленник вторгается на удаленной машине и получает локальный доступ к машине жертвы. Например. подбор пароля соответствующих функций: Функции сетевого уровня – “duration of connection” and “service requested”, а также функции уровня хоста - - “number of failed login attempts”

Общее количество данных: 144625(100%)

Данных для обучения: 125973(87.1%)

Данных для проверки: 18625(12.9%)

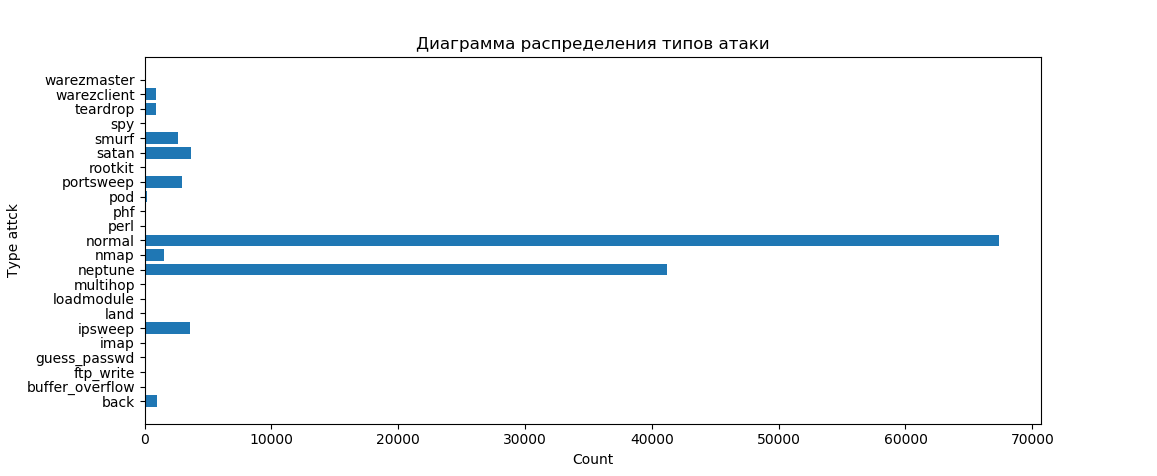
Всего признаков 42(100%)

Целочисленных признаков:23(54.8%)

Вещественных признаков:15(35.7%)

Символьных признаков: 4(9.5%)

Целевая переменная: Тип атаки (41)



Поговорим немного про категориальные признаки

Как видно из данных имеется 3 категориальных признака. Классификаторы не умеют работать с буквами, им нужны цифры.

Для решения этой проблемы существует несколько вариантов решений.

Поговорим о паре из них.

Самым простым методом отображения категориальных признаков в числовые является LabelEncoder который отображает признаки в значения на числовой оси

На примере наших данных посмотрим какие будут изменения в признаке ‘protocol\_type’

"icmp --> 0"

"tcp --> 1"

"udp --> 2"

Очевидны проблемы этого метода

Теперь tcp + tcp = udp (1 + 1 = 2)

Также можно подумать, что задано отношение порядка, хотя его не может быть

На помощь приходит метод OneHotEncoding который переводит категориальные признаки в бинарный вектор где «1» стоит на месте где встречается признак

В нашем примере OneHotEncoding сработает следующим образом:

|  |  |  |  |
| --- | --- | --- | --- |
| protocol\_type | icmp | tcp | udp |
| icmp | 1 | 0 | 0 |
| tcp | 0 | 1 | 0 |
| udp | 0 | 0 | 1 |

Тогда на выход мы получим

icmp 🡪 [1,0,0]

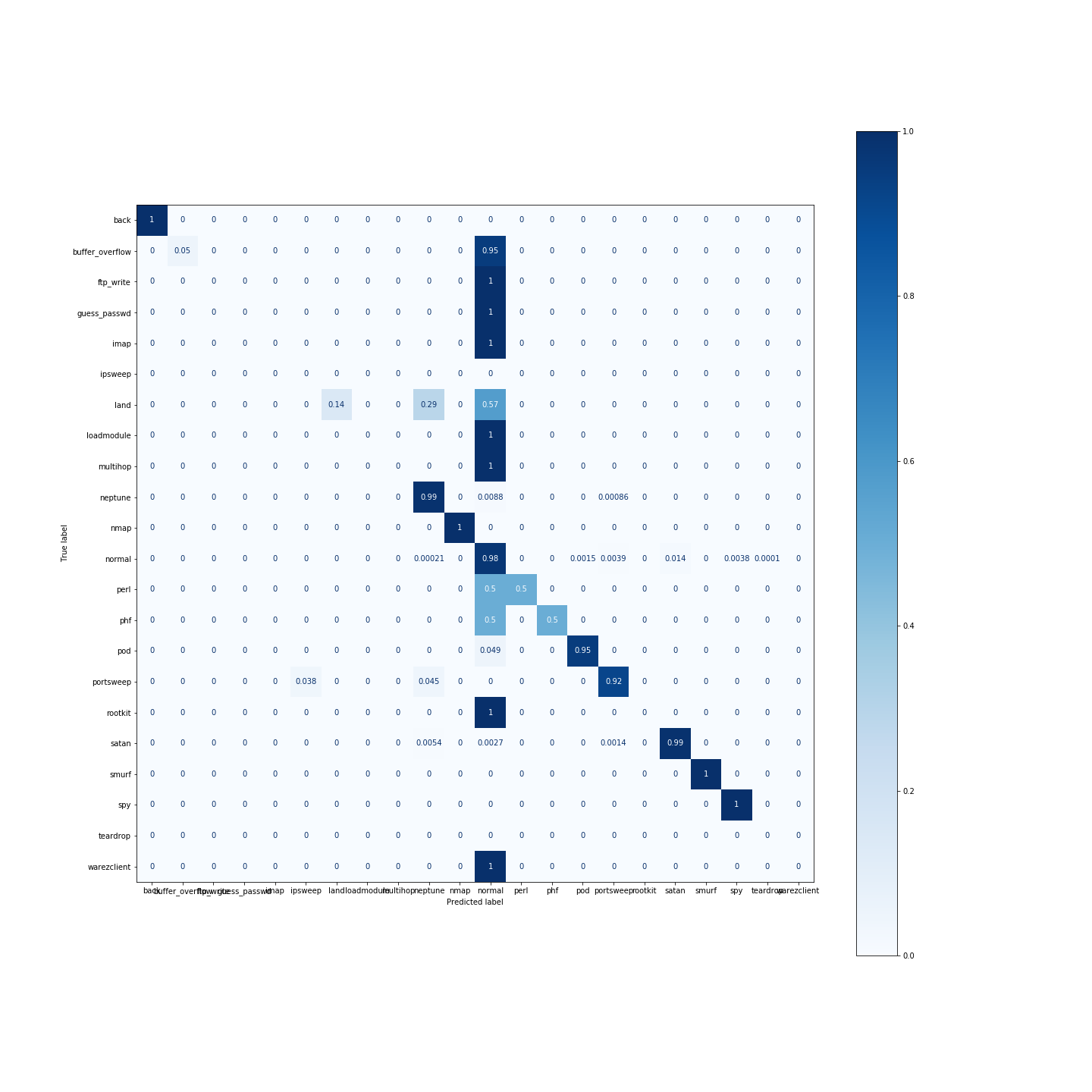
tcp 🡪[0,1,0]

udp 🡪[0,0,1]

Хотя этот метод устранаяет проблему иерархии данных, однако увеличивает вычислительную сложность т.к. добавляет дополнительные столбцы в данные (по количеству уникальных значений признака)

Feature Engineering

Матрицу ошибок будем строить по RandomForestClassifier

Из матрицы ошибок, видно, что многие атаки классифицируются неправильно

Возьмем, например, атаку buffer\_overflow и подумаем, как можно сделать его различимым.

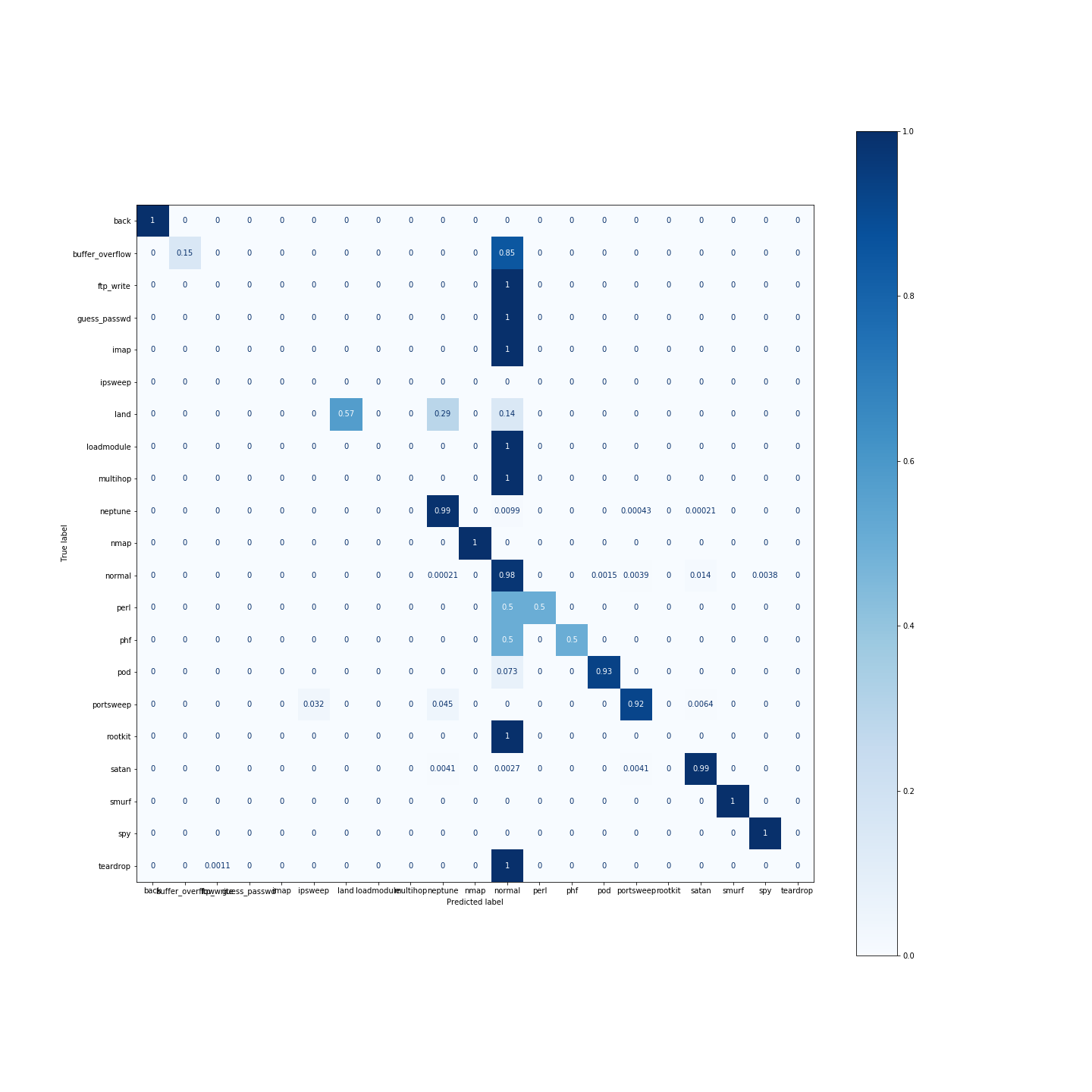
Немного погуглив можно найти статью <https://owasp.org/wwwcommunity/vulnerabilities/Buffer_Overflow>

В которой сказано, что buffer\_overflow атака использует переполнение размера буфера.

В наших данных есть несколько признаков связанных с количеством байтов (src\_bytes, dst\_bytes).Тогда составим признак и посмотрим, что получится: Новый признак = (dst\_bytes – src\_bytes)

Посмотрим на качество классификации и матрицу ошибок

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | DecisionTreeClassifier | RandomForestClassifier |
| Обучающая выборка | base | 0.9967453859892836 | 0.9978567176026989 |
| new | 0.9969041476483429 | 0.9979360984322286 |
| Тестовая выборка | base | 0.8327257130602617 | 0.8639288012009436 |
| new | 0.8567981985846022 | 0.8639288012009436 |



Из матрицы видно, что повысилась точность классификации данной атаки на 0.1

Еще начали лучше отличать атаку land

Далее посмотрев на ftp\_write и немного погуглив, узнаем что это атака связана с ftp соединением.

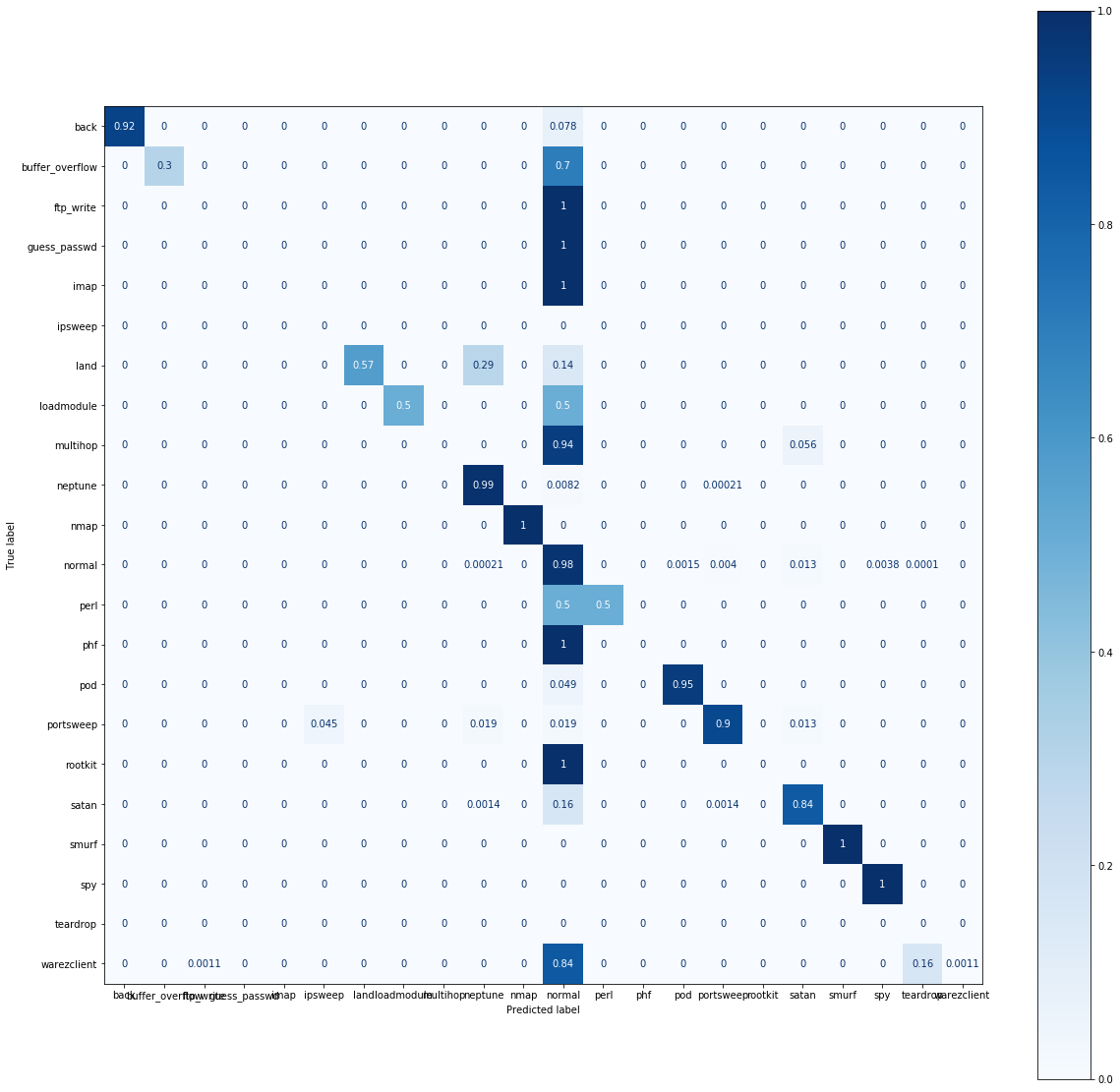
Тогда попробуем увеличить вес признака связанного с ftp

А у нас как раз такие данные имеются , попробуем составить новый признак

Новый признак = num\_outbound\_cmds\* 1000

Посмотрим на качество классификации и матрицу ошибок

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | DecisionTreeClassifier | RandomForestClassifier |
| Обучающая выборка | base | 0.9967453859892836 | 0.9978567176026989 |
| new | 0.9968644572335781 | 0.9977773367731693 |
| Тестовая выборка | base | 0.8327257130602617 | 0.8639288012009436 |
| new | 0.8197512331117306 | 0.8575487883336907 |



Из матрицы видно. Что ftp\_write не изменился, однако улучшилось качество атаки buffer\_owerflow, land. А самое интересное, мы начали отличать атаку loadmodule от normal в половине случаев

Рассмотрим различные модели и параметры

|  |  |  |
| --- | --- | --- |
|  | No tuned params | |
|  | Train | Test |
| RandomForest | 0.9977376463584045 | 0.8639288012009436 |
| GradientBoosting | 0.9955180789839253 | 0.8587282865108299 |
| SVC | 0.9834094066282992 | 0.8303667167059833 |

|  |  |  |
| --- | --- | --- |
|  | GreedSearch | |
|  | Train | Test |
| RandomForest(n\_estimators=200,max\_depth=10,max\_features=50) | 0.9965072435006945 | 0.8619450997212095 |
| GradientBoosting(n\_estimators=10,max\_depth=6,max\_features=50) | 0.9951974598134551 | 0.8619450997212095 |
| SVC(C=1,gamma=0.1,kernel=’sigmoid’) | 0.9621750347291129 | 0.8323504181857173 |

|  |  |  |
| --- | --- | --- |
|  | Random Search | |
|  | Train | Test |
| RandomForest(n\_estimators=100,max\_depth=10,max\_features=50) | 0.9965469339154595 | 0.8632318250053613 |
| GradientBoosting(n\_estimators=175,max\_depth=10,max\_features=75) | 0.9974598134550506 | 0.8529380227321467 |
| SVC(C=1,gamma=0.9,kernel=’poly’) | 0.9959912681087517 | 0.8266673815140467 |