Пояснительная записка.

Здравствуйте, меня зовут Процкая Елизавета, я студентка группы 15-В-1 ,и я хочу рассказать о программной системе анализа сетевого трафика. Данная задача приобретает все большую актуальность в связи с развитием и внедрением новых сетевых технологий и, как следствие, увеличением объема данных, передаваемых по сети, а также появлением большого количества новых сетевых протоколов прикладного уровня.

2. Данная работа посвящена описанию реализации программной системы для анализа сетевого трафика через решение задачи определения протокола прикладного уровня и использования алгоритма машинного обучения. (перечисление задач)

3. Начнем мы с рассмотрения различных подходов к решению. Для решения задачи классификации трафика создано множество алгоритмов. Один из вариантов классификации этих подходов приведен на слайде.

Существуют два основных метода классификации трафика:

- Классификация на основе статистического метода. (Statistical Analysis)
- Классификация на основе блоков данных (Payload-Based Classification).

При классификации на основе блоков данных выделяют универсальный подход к классификации трафика и метод глубокого анализа пакетов (DPI).В данной программной системе используется метод DPI(deep packet inspection) так как он позволяет осуществить более точную классификацию.

Системы глубокого анализа трафика позволяют классифицировать те приложения и протоколы, которые невозможно определить на IP-адресах и портах, например URL внутри пакета, содержимое сообщений мессенджеров, голосовой трафик Skype, p2p-пакеты BitTorrent. То есть DPI анализирует не только заголовки пакетов, но и полное содержимое трафика на уровнях модели OSI со второго и выше.

Применение метода глубокого анализа пакетов при современных объемах трафика требует больших вычислительных затрат, и соответственно получает низкую скорость проведения анализа трафика. Поэтому было принято решение использовать альтернативный способ решения одной из главных задач DPI – определения протокола прикладного уровня на основе применения алгоритмов машинного обучения.

Этот метод будет иметь более высокую скорость анализа, так как он основан на малом количестве информации, не требует оценивать и запоминать содержимое IP-пакетов, не сверяясь со списком широко известных портов, и не глядя в полезную нагрузку пакетов

4. Итак, наша система имеет следующую информационную модель (слайд 3). Захват трафика производится с помощью программы Wireshark– это программа, которая поддерживает разбор большого количества различных сетевых протоколов, а также предоставляет возможность сортировки и фильтрации трафика. Весь захваченный трафик мы делим на обучающую и тестовую выборки. На первом этапе данные обущающей выборки подвергаются предварительной обработке. А затем по ним формируются признаки и заносятся в базу признаковых описаний, после чего соответственно получаем обученную модель. Далее остальной объем трафика так же подвергается обработке и на основании ранее обученной модели происходит классификация.

5. Этап предварительной обработки заключается в устранении следующих недостатков: (слайд).

6. Идея, которая лежит в основе предлагаемого метода, заключается в том, что разные приложения, пользующиеся разными протоколами, также генерируют потоки транспортного уровня с разными статистическими характеристиками. Если аккуратно и ёмко определить набор статистических метрик потока, то по значениям этих метрик можно будет с высокой точностью предсказывать, какое приложение сгенерировало данный поток, и, соответственно, какой протокол прикладного уровня этим потоком переносится. В данной программной системе все используемые статистические характеристики потока будут основаны на этих четырех рядах чисел. Эти 4 ряда чисел достаточно хорошо характеризуют поток данных, и на их основе можно с высокой точностью предсказать протокол прикладного уровня.

7. При выборе алгоритма машинного обучения, нельзя однозначно точно заранее определить какой классификатор для поставленной задачи будет лучшим, на эффективность алгоритмов влияет множество факторов, например, объем и структура исследуемых данных. Следовательно, необходимо проверять эффективность каждого в каждом конкретном случае на тестовом наборе данных и затем выбирать лучший

вариант. В решение этой задачи и помогает Microsoft Azure ML— это облачная служба аналитики, которая позволяет быстро создавать и развертывать прогнозные модели в качестве решений аналитики. Общая схема проведенного эксперимента изображена на слайде.

Data.csv- набор ранее собранных и преобразованных данных.

Рис.6. Табличное представление собраных данных

Split Data – разделяет данные на две части в пропорции 0.75 –для обучающей выборки, 0.25 –для тестовой выборки (В схеме использовались две Split Data для наглядности).

Train Model – обучает модель, на вход принимает один из представленных классификаторов и обучающую выборку, а выдает обученную модель. Так же необходимо указывать параметр, который будет предсказываться – в данной задаче это proto, то есть протокол.

Tune model Hyperparametrs – у каждого классификатора есть параметры настройки. Если для каждого параметра одно значение, используется Train Model, а если несколько, то данный элемент, который прогоняет все комбинации параметров, находит лучший и обучается по нему модель.

Score Model – предсказывает значение, на вход принимает обученную модель и тестовую выборку, иа на выход выдает результат.

Evaluate Model – отображается полученные данные от Score model.

8. Хотя из результатов исследования видно, что лучшие значения имеют логическая регрессия и нейронная сеть, в разрабатываемой программной системе будет использоваться алгоритм из Multiclass Decision Forests, а именно Random Forest. Так как он показал достаточно высокую точность классификации, он проще в реализации и потому что из-за достаточно большого количества признаков, нам нужен алгоритм, который слабо чувствителен к шумам и корреляции метрик.

9. Данный алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт не очень высокое качество классификации, но за счёт их большого количества результат получается хорошим.

На слайде приведён алгоритм обучения классификатора.

Дан обучающий набор размерности N, размерность множества признаков M и параметр m (обычно m = sqrt(M)).

1) Генерируется случайная выборка с повторением размером N из обучающего набора.

2) Строится дерево принятия решений, которое классифицирует примеры из данной выборки. В ходе создания очередного узла дерева выбирается признак, на основе которого производится разбиение, не из всех M признаков, а лишь из m случайно 6 выбранных. Выбор наилучшего из этих m признаков может осуществляться различными способами.

3) Строится дерево до полного исчерпания выборки,

Описанные выше шаги, повторяются для каждого дерева. Классификация объектов проводится путём голосования: каждое дерево комитета относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев. Оптимальное число деревьев подбирается таким образом, чтобы минимизировать ошибку классификатора на тестовой выборке.

10. Признаковые описания классифицируют собранный трафик по выбранным протоколам. Исследуется таблица характеристик классифицированного трафика. Решение о точности классификации принимается на основе числа правильных классификаций трафика и числа ошибок.

11. С помощью программы Wireshark произведен захват 10гб трафика и отобраны следующие виды DNS, BitTorrent, HTTP(S), SSL, Skype.

Для выделения этих видов среди прочего трафика использована библиотека nDPI, осуществляющая глубокий анализ пакетов.

Для запуска программный продукта необходим интерпретатор Python версии 2.7.x с установленными библиотеками PyQt4, dpkt, numpy, sklearn и pandas.

12. По результатам классификации получаем две таблицы.

Первой представлена таблица точности и полноты предсказаний по каждому классу. Точность для класса К – это доля предсказаний вида «объект Х принадлежит классу К», которые оказались верными. Полнота для класса К – количество объектов Х, которые распознаны классификатором как принадлежащие классу К, делённое на общее количество принадлежащих классу К объектов. Ф-мера – среднее гармоническое полноты и точности.

И соответственно таблица результатов.
Здесь в каждой ячейке указано число, которое означает количество случаев, когда поток трафика, принадлежащий указанному в заголовке строки классу, был классифицирован как принадлежащий указанному в заголовке столбца классу. То есть, числа на главной диагонали – правильные классификации, числа вне её – разного рода ошибки.
Как мы можем видеть, что данный метод с достаточно большой точностью классифицирует трафик.

13. Результатом работы разрабатываемой системы является осуществление захвата трафика в реальном времени и предсказывание его протокола прикладного уровня