|  |
| --- |
| **Методы Data Mining в задачах обнаружения нарушений** |
| План:  1. Нормальное состояние и аномалии.  2. Data Mining – основные понятия.  3. Обработка данных TCPdump.  Литература. |
| **1. Нормальное состояние и аномалии**  На Рисунке 1 показана разница в числе TCP-пакетов с установленными syn и synack флагами (данные приведены для tcpdump\_class.log, размер окна ~10 сек ). На фоне постоянного или относительно медленного изменения можно видеть три резких скачка, помеченных на рисунке цифрами 1, 2 и 3 соответственно. Эти скачки расположены в области пакетов с номерами:   1. 11600 – 15000 2. 30000 – 30200 3. 45340 - 46850   На рисунке можно видеть и другие скачки, но остановимся на этих трёх, самых заметных.  Используя программу Wireshark () проанализируем эти скачки и попытаемся выяснить причину их появления.  ***1-й скачок*** (~ 11600 – 15000): Вызван тем, что узел 213.169.64.189 перестал отвечать на запросы syn. До и после указанного интервала он отвечает нормально. Вывод: "Упал" сервер.  ***2-й скачок*** (~ 30000 – 30200): Вывод: syn - сканирование.  ***3-й скачок*** (~ 45340 - 46850): Узел 61.8.97.153 посылает запросы к 212.193.209.129 на порты 135 и 5000. (Посмотреть в INET назначение этих портов). Вывод: Попытка вызвать "Троян".  Скачки – аномалии, остальное – нормальное состояние.    Рисунок 1  **2. Data Mining – основные понятия**  Характеристики (Признаки).  Пространство характеристик.  Редукция размерности (многомерное шкалирование, метод главных компонент, факторный анализ)  Классификация и кластеризация. |

|  |
| --- |
| **3. Обработка данных TCPdump**  Характеристики:   1. Разница между syn и synack -пакетами в окне. 2. Временной интервал между двумя последними пакетами с одного узла. 3. Наибольшее значение числа пакетов с одного узла в окне. 4. Среднее значение по окну разницы идентификаторов двух последних пакетов. 5. Среднее значение по окну размера пакета.     Рисунок 2  На Рисунке 2 приведен результат редукции размерности пространства характеристик к двумерному случаю (использовано многомерное шкалирование). Каждая точка на рисунке соответствует одному событию (совокупность пакетов в окне ~ 10 сек) со значениями перечисленных 5-ти характеристик. Красным цветом обозначены события с аномально высоким значением разницы числа syn и synack пакетов, соответствующих пикам на графике Рисунка 1.  На Рисунке 3 приведён пример syn-сканирования.  Рисунок 3  Расположение точек в пространстве характеристик (многомерное шкалирование) для случая syn-сканирования (данные Рисунка 3).    Рисунок 4  Те же данные syn-сканирования, но без учёта syn-ack пакетов. Тем не менее выделяется аномалия!!!    Рисунок 5 |

**Редукция размерности пространства характеристик**

**Метод главных компонент**

D – размерность пространства характеристик, – число объектов.

Характеристики центрированы и нормированы:

Попарные корреляции:

Или (без нормирования и центрирования):

Корреляционная матрица составлена из попарных корреляций: . (Заметим, что это самосопряжённая матрица.)

Найдём собственные вектора и собственные значения матрицы :

Матрицу можно представить в виде:

Если упорядочить собственные вектора в порядке убывания собственных значений, то основной "разброс" в корреляционной матрице будут обеспечивать несколько первых членов суммы.

При проекции множества точек в пространстве характеристик на плоскость , эту плоскость мы выбираем по первым двум собственным векторам.

Результат такой проекции для множества событий соответствующих Рисунку 1 приведён на Рисунке 6:

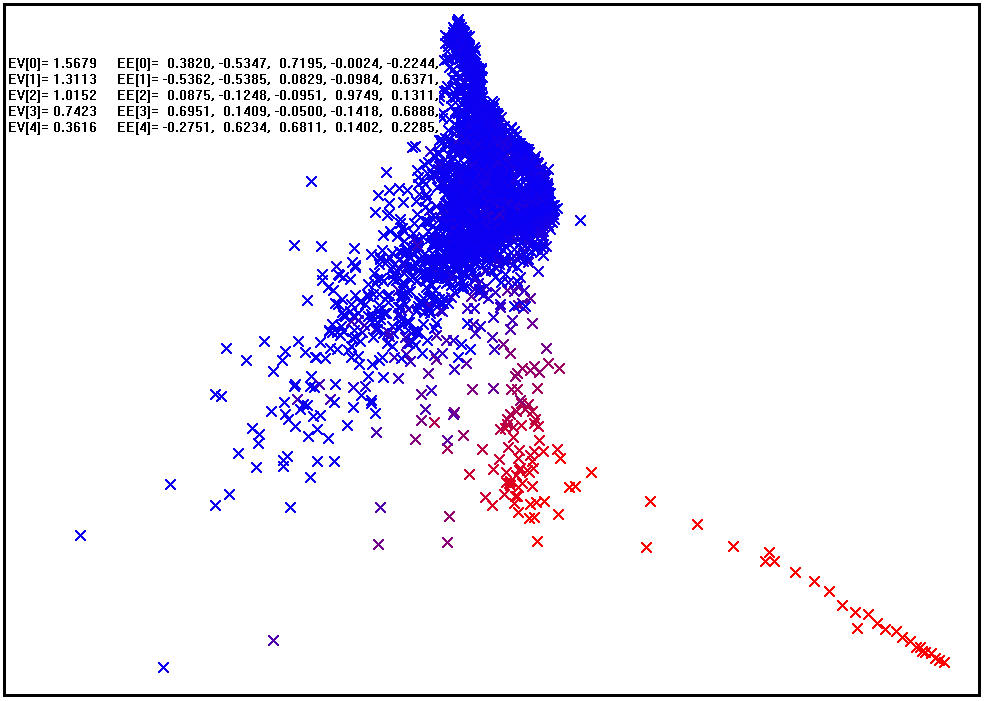


Рисунок 6

**Многомерное шкалирование**

Пусть – попарные расстояния между -й и -й точками в пространстве характеристик (введена метрика, например Евклидова). Отобразим точки на некоторую двумерную плоскость и пусть - расстояние между соответствующими точками на плоскости.

Минимизируем значение стресс-функции:

Для этого

В случае Евклидовой метрики

Мы получаем

cистему уравнений для координат точек на плоскости. (Эту систему лучше решать методом итераций, в качестве начального приближения можно использовать результат метода главных компонент).

Для сравнения приведём результаты этих двух способов:

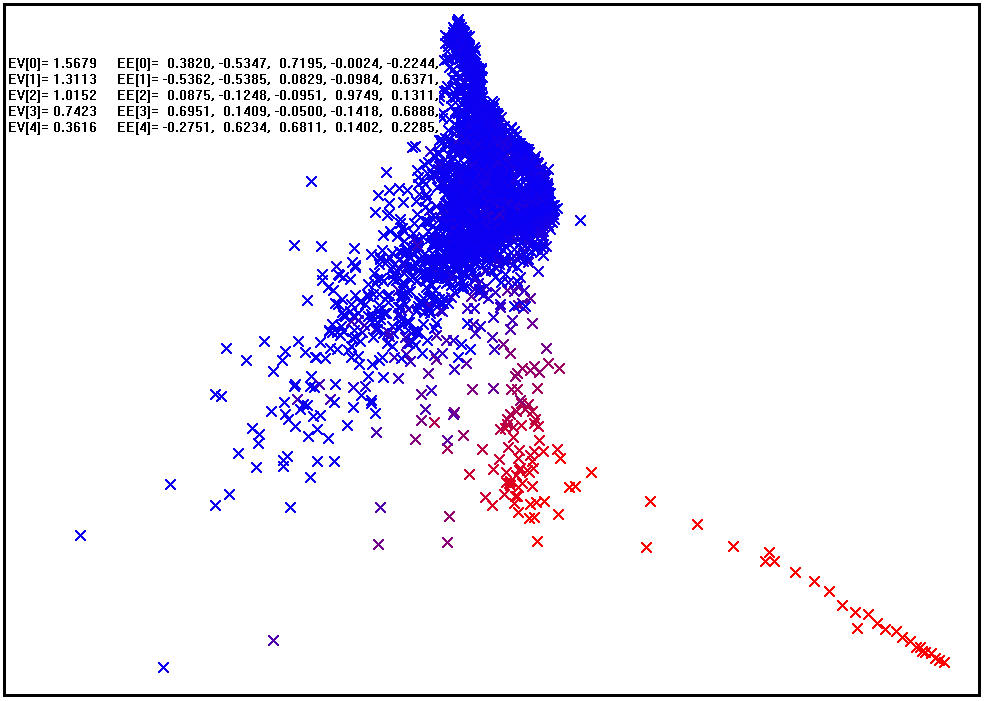
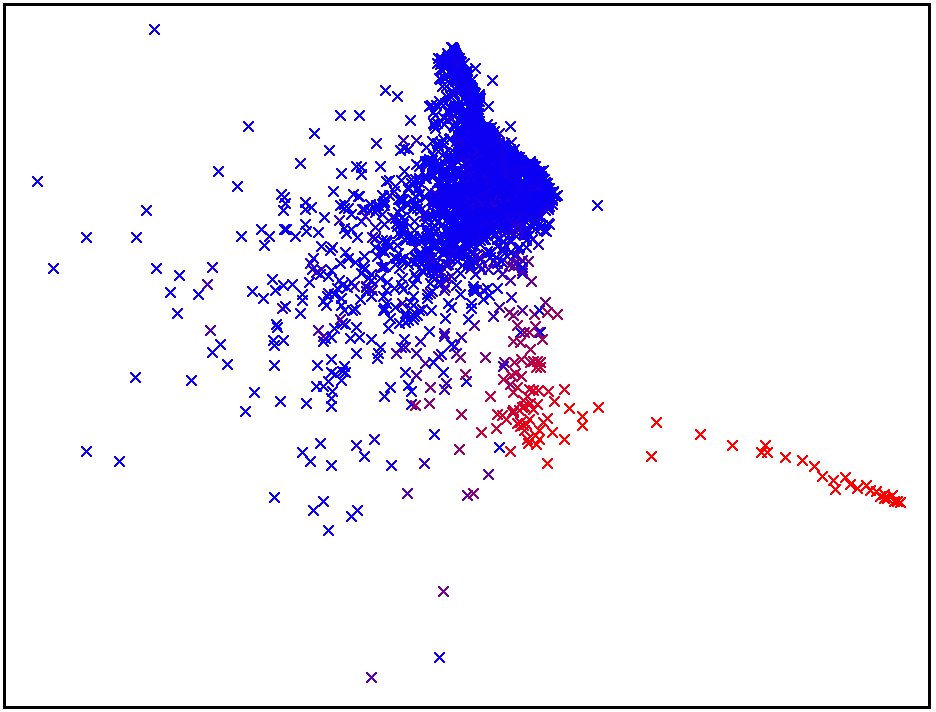
Слева – многомерное шкалирование, справа – метод главных компонент.

Рисунок 7

**Факторный анализ**

**Кластеризация данных**

Рассматриваемые методы являются итерационными методами, каждая итерация состоит из двух шагов:

1. На основе характеристик кластера (центр, радиус, …) выявляются точки ему принадлежащие.

2. По набору точек входящих в кластер уточняются его характеристики.

**Метод K-means**

Рассматриваем двумерный случай. Каждый кластер описывается структурой:

typedef struct \_CLUST {

int N; // число точек в кластере

double X[2]; // координаты центра

double S[2][2]; // ковариационная матрица

double SI[2][2]; // обратная ковариационная матрица

double sx; // sum (x)

double sy; // sum (y)

double sxx; // sum (x\*x)

double syy; // sum (y\*y)

double sxy; // sum (x\*y)

} CLUST;

Кластер имеет форму эллипса и определяется следующими характеристиками:

1. N - число точек в кластере.
2. Координаты его центра: .
3. Матрица ковариаций: .
4. Обратная ковариационная матрица: , . Матрица определяет форму и размер кластера, задаёт расстояние от точки до центра кластера. Метрика Махаланобиса: .

На первом шаге итерации принадлежность точки кластеру определяется из соотношения:

Новое наблюдение будет аномальным, если выходит за пределы доверительного интервала , где - среднее значение, - стандартное отклонение, - некоторый параметр. , . Значение параметра зависит от задаваемой вероятности выхода события за пределы доверительного интервала. Для любой функции распределения справедливо неравенство Чебышева: . Это означает, что для любого вида распределения величины и при значении параметра вероятность выйти за пределы доверительного интервала не превышает .

На втором шаге итерации вычисляются центр кластера и матрица ковариаций через вспомогательные параметры:

Конкретно:

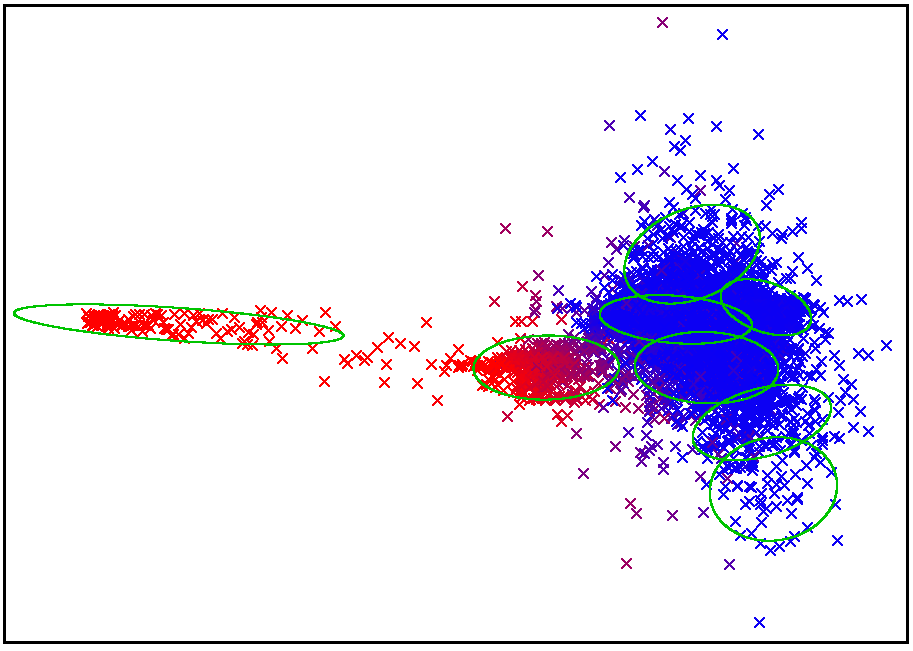
и затем обратную матрицу .

Так как число кластеров неизвестно, мы используем адаптивную версию алгоритма K-means:

1. Инициализируем пустой набор кластеров.
2. Для каждого новой точки ищем ближайший кластер Если такой кластер найден, то включаем в состав этого кластера и пересчитываем характеристики кластера с учётом новой точки.
3. Если подходящий кластер не найден, то создаём новый кластер с центром в точке с «затравочным» радиусом и включаем его в набор кластеров.
4. Если центр какого-либо кластера оказывается внутри другого, то эти два кластера объединяем в один кластер.
5. Возвращаемся к шагу 2 для обработки следующей точки.

После обработки всех точек удаляем кластеры с небольшим числом точек.

Результат приведён на рисунке:



Теперь, используя положения и размеры кластеров, используем EM итерации (…). Результат на рисунке:

