# АНАЛИЗ IP-ТРАФИКА МЕТОДАМИ DATA MINING. ПРОБЛЕМА КЛАССИФИКАЦИИ

Н. Г. Щербакова

Институт вычислительной математики и математической геофизики,
630090, Новосибирск, Россия

УДК 681.324

Предложен подход к решению задачи классификации IP-трафика, основанный на методах интеллектуального анализа данных. На основе статистических параметров потоков, извлекаемых из характеристик, не зависящих от полезной нагрузки IP-пакетов, проводится идентификация сетевых приложений. Для решения задачи применяется анализ данных, обеспечивающий автоматическое выявление скрытых закономерностей. Рассмотрен ряд алгоритмов классификации и поиска. Проведен их сравнительный анализ.

**Ключевые слова:** классификация IP-трафика, P2P-трафик, методы машинного обучения, эффективность и стабильность алгоритмов классификации.

The methodology of IP traffic classification based on the intellectual data analysis is introduced. The identification of network applications is based on statistical flow characteristics derived from payload-independent features. Data mining techniques is used for automatic extraction of hidden patterns. The set of classification, clustering and association rules extracting algorithms are examined. The comparison of the algorithms is presented.

**Key words**: IP traffic classification, Peer-to-Peer traffic, machine learning technique, algorithms efficiency and stability.

**1. Проблема и исследуемые данные.** В настоящей работе рассматривается проблема анализа IP-трафика с использованием методов Data Mining, к числу которых относятся классификация — разделение объектов на заданные группы (классы) согласно характеристикам объектов; регрессия — поиск функции, моделирующей множество изучаемых объектов с наименьшей ошибкой; кластеризация — поиск независимых групп объектов, кластеров и их характеристик (группы заранее не предопределены); поиск ассоциативных правил — характерных зависимостей между объектами или событиями.

Формально методы поиска закономерностей можно сформулировать следующим образом [1].

Задачи классификации и регрессии. Имеется множество исследуемых объектов $X = \{x_1, x_2, \ldots, x_n\}$. Каждый объект характеризуется набором переменных (атрибутов) $X_j = \{a_1, a_2, \ldots, a_m, y\}$, где $a_i$ — наблюдаемые переменные, значения которых известны; $y$ — зависимая переменная, значение которой нужно определить. При этом каждая переменная $a_i$ принимает значение из некоторого множества $A_i = \{a_{i1}, a_{i2}, \ldots\}$. Наблюдаемые переменные часто называются признаками или атрибутами. Если множество $C = \{c_1, c_2, \ldots, c_k\}$ значений переменной $y$ конечно, то задача называется задачей классификации. Если переменная $y$ принимает значения на множестве действительных чисел $\mathbb{R}$, то задача называется задачей регрессии.

Задача кластеризации. Имеется множество исследуемых объектов $X = \{x_1, x_2, \ldots, x_n\}$. Каждый объект характеризуется набором переменных $X_j = \{a_1, a_2, \ldots, a_m\}$. Каждая переменная $a_i$ принимает значение из некоторого множества $A_i = \{a_{i1}, a_{i2}, \ldots\}$. Задача кластеризации состоит в построении множества $C = \{c_1, c_2, \ldots, c_k\}$, где $c_i$ — кластер, содержащий сходные объекты из множества $X$, относительно введенной меры близости $d(x_j, x_p)$, называемой расстоянием, т. е. $c_m = \{x_j, x_p | x_j \in X, x_p \in X \& d(x_j x_p) < \sigma\}$, где $\sigma$ — величина, определяющая максимальное расстояние, на котором могут находиться объекты одного кластера.

Задача поиска ассоциативных правил. Имеется набор исходных элементов $I = \{i_1, i_2, \ldots, i_n\}$, а также набор объектов $D = \{d_1, d_2, \ldots, d_m\}$. Каждый объект является подмножеством множества $I$ ($d_i \subseteq I$). В соответствии с терминологией, относящейся к базам данных, $d_i$ называется транзакцией, а $D$ — базой данных. Правило — это импликация вида $X \Rightarrow Y$ , где $X, Y \subseteq D$ и $X \cap Y = \emptyset$. При этом для выявления наиболее правдоподобных правил, отражающих часто встречающиеся зависимости между транзакциями в базе данных, вводятся две метрики. Поддержка набора $X$, обозначаемая как $\mathrm{supp}(X)$, — это пропорция набора $X$ относительно всего множества $D$. Поддержка правила $\mathrm{supp}\,(X \Rightarrow Y) = \mathrm{supp}\,(X \cup Y)$. Доверие к правилу определяется по формуле $\mathrm{conf}(X \Rightarrow Y) = \mathrm{supp}\,(X \cup Y)/\mathrm{supp}\,(X)$. Чем больше значения поддержки и доверия, тем более точно правило отражает зависимости.

С алгоритмической точки зрения классификация или кластеризация — это функция $f : X \to C$, которая каждому объекту $x_i \in X$ ставит в соответствие метку $c_j \in C$. В задаче классификации множество $C$ определено заранее, в задаче кластеризации заранее не определено не только множество $C$, но и его размерность.

Для реализации методов используются как базовые статистические алгоритмы, так и генетические алгоритмы, нейронные сети, алгоритмы из области машинного обучения. Машинное обучение (machine learning [2]) — научная дисциплина, изучающая методы построения алгоритмов, способных обучаться на основе прецедентов. В общем виде постановку задачи можно представить следующим образом. Имеется множество объектов (ситуаций) и множество возможных ответов (откликов, реакций). Существует некоторая зависимость между ответами и объектами, но она не известна. Известна только конечная совокупность прецедентов — пар "объект — ответ", называемая обучающей выборкой. На основе этих данных требуется восстановить зависимость, т. е. построить алгоритм, способный для любого объекта выдать достаточно точный ответ.

По способам обучения рассматриваемые алгоритмы классифицируются следующим образом: контролируемое (supervised) обучение — обучение на помеченных данных, когда для каждого прецедента задается отображение входные данные → желаемое решение и требуется изучить функцию отображения, например с целью дальнейшей дифференциации и классификации любых входных данных; неконтролируемое (unsupervised) обучение — обучение, когда помеченные данные не предоставляются и требуется сгруппировать объекты в группы (кластеры) на основании "близости" объектов (относительно некоторой меры близости); частично контролируемое (semi-supervised) обучение — обучение, когда изучение функции отображения осуществляется на комбинации помеченных и непомеченных данных.

Для определения эффективности алгоритма используются следующие метрики: *FP* (false positive) — доля трафика, приписанного к классу $X$, но не принадлежащего к $X$ по отношению к мощности $X$; *FN* (false negative) — доля трафика, принадлежащего к классу $X$, но не приписанного к классу $X$; правильность (accuracy) — доля правильно классифицированных

единиц по отношению ко всем классифицированным единицам, т. е. $(all\text{-}FP\text{-}FN)/all$. Точность (precision) — пропорция правильно классифицированных единиц ($TP$) относительно полученного класса: $TP/(TP{+}FP)$; полнота (recall) / доверие (trust) — пропорция правильно классифицированных единиц относительно реального класса: $TP/(TP{+}FN)$.

Исходными данными для исследования являются последовательности IP-пакетов, собранных в точках наблюдения. Единицей рассмотрения является поток. Последовательность IP-пакетов может быть двунаправленной или однонаправленной последовательностью пакетов между двумя IP-адресами, полной TCP-сессией или однонаправленной последовательностью IP-пакетов, определяемой на основе пяти полей заголовка

$$<src\_ip,\ src\_port,\ dst\_ip,\ dst\_port,\ protocol>$$

и правил формирования, по которым определяется завершение потока (обычно тайм-аут или флаг "END" в заголовке пакета). Здесь $src\_ip$ — IP-адрес источника; $src\_port$ — порт источника; $dst\_ip$ — IP-адрес назначения; $dst\_port$ — порт назначения, $protocol$ — транспортный протокол. Как правило, в качестве протоколов транспортного уровня используются TCP и UDP.

Фиксируется набор переменных (атрибутов), основанных на статистических характеристиках, таких как размер пакетов или интервалы между пакетами, и характеристиках, извлекаемых из заголовков пакетов, таких как размер TCP-сегментов или количество повторных передач. Потоку ставится в соответствие набор значений атрибутов, согласно которым проводится классификация.

При описании алгоритмов и статистических характеристик используются термины теории вероятностей и математической статистики, определения которых приведены в [3], а также в статьях интернет-энциклопедий.

**2. Классификация IP-трафика на основе статистической модели поведения протокола прикладного уровня.** В работе [4] исследуется поведение протоколов прикладного уровня. Изучаются однонаправленные TCP-потоки, направленные от клиента к серверу и от сервера к клиенту и относящиеся к сессии прикладного уровня. Для каждого потока рассматривается упорядоченная последовательность пар $P_i = (s_i, \Delta t_i)$, где $s_i$ — размер $i$-го пакета, байт; $\Delta t_i$ — интервал между $i$-м и предыдущим пакетами. Эти характеристики выбраны в предположении, что механизмы генерации потоков прикладного уровня используют уникальное соотношение значений этих параметров. Статистическая модель, построенная на основе проверки данных характеристик, называется protocol fingerprinting — "отпечатки пальцев протокола".

Сначала изучаются тренировочные образцы потоков, соответствующих выбранным приложениям. В настоящей работе приводятся данные о протоколах HTTP, SMTP, POP3. Потоки делятся на две группы: клиентский трафик и трафик, инициированный сервером. Для каждой группы каждого рассматриваемого протокола строится вектор функций плотности вероятности **PDF** (probability density functions). Каждая компонента вектора $PDF_i$ — эмпирически найденная совместная плотность (по потокам) вероятности того, что для $i$-х пар размер пакета равен $s$, интервал между пакетами равен $\Delta t$ при длине потока не меньше $i{+}1$. Длина $L$ вектора **PDF** зависит от размера наиболее длинного потока, зафиксированного в течение тренировочной фазы.

Диапазон длин IP-пакетов — дискретная величина, зависящая от сетевого интерфейса, на котором собираются данные. Например, для линии связи Ethernet значение этой величины находится в диапазоне [40, 1500]. Минимум для интервала между пакетами зависит

от скорости линии и разрешимости таймера на интерфейсе, в данном случае рассматривается диапазон от $10^{-7}$ до $10^{3}$ с с шагом $10^{-2}$. Каждая функция $\mathbf{PDF}_i$ представляется в виде матрицы, каждый элемент которой отображает вероятность того, что $i$-й пакет характеризуется соответствующими значениями $s_i$ и $\Delta t_i$. Поскольку оценка $\mathbf{PDF}_i$ проводится на основе анализа тренировочных данных, матрица обрабатывается фильтром Гаусса [5], используемым для подавления уровня "шума", например при задержке пакета вследствие отказа сети. Затем матрица масштабируется, чтобы сумма вероятностей составляла 1. Два отфильтрованных вектора **PDF** (клиента и сервера) для протокола $j$ составляют protocol fingerprint $\mathbf{M}^j$.

Процесс классификации представляет собой соотнесение характеристик исследуемого потока $F$ со всеми известными $\mathbf{M}^j$. В данной работе приводятся два алгоритма классификации: probability product algorithm и anomaly score algorithm.

*Probability product algorithm.* Сначала последовательность пар $P_i$, соответствующая исследуемому потоку длиной $N$, преобразуется в последовательность $Z_i^j$ длиной $L$ для каждого изученного протокола $j$. Если $N \le L$, то

$$Z_i^j = \begin{cases} \mathbf{M}_i^j P^j(N \ge i), & N \ge i, \\ P^j(N < i), & N < i. \end{cases}$$

Здесь $P^j(N \ge i)$ — вероятность того, что потоки протокола $j$ имеют длину не меньше $i$; $P^j(N < i)$ — вероятность того, что потоки протокола $j$ имеют длину меньше $i$. Если $N > L$, то остальные коэффициенты отбрасываются. Вычисляется произведение коэффициентов

$$V(F, \mathbf{M}^j) = \prod_{i=1}^{L} Z_i^j,$$

которое затем сравнивается с пороговым значением $T$. Если величина для некоторого протокола $j$ больше порогового значения, то поток помечается этим протоколом, в противном случае он считается нераспознанным.

*Anomaly score algorithm.* Алгоритм использует вектор аномальности $\mathbf{A}$ для принятия решения, соответствует ли поток хотя бы одному "отпечатку" протокола. Компонента $A_i$ определяется для протокола $j$ следующим образом:

$$A_i(P_i, \mathbf{M}_i^j) = 1/\max(\varepsilon, \mathbf{M}_i^j(P_i)),$$

где $\varepsilon$ — некоторая малая величина, гарантирующая определенность выражения. Алгоритм вычисляет выражение для каждой пары $(s_i, \Delta t_i)$ потока, после рассмотрения $n$ пар потока $F$ определяется счетчик аномальности

$$S_n(F, \mathbf{M}^j) = \left( \left( \sum_{i=1}^{n} A_i\left(P_i, \mathbf{M}_i^j\right)/n \right) - A_{\min} \right) \Big/ (A_{\max} - A_{\min}),$$

где $n$ — минимум между количеством пар потока $F$ и $L$; $A_{\max}, A_{\min}$ — разрешенные экстремальные значения для $A$, например $\varepsilon^{-1}$ и 1. Счетчик аномальности нормализуется с помощью вычисленного заранее порога $T_n^j$. Если порог не превышен, поток помечается протоколом, имеющим минимальное значение $S_n$. Порог $T_n^j$ определяется через математическое ожидание и отклонение счетчиков аномальности для потоков, используемых при построении вектора $\mathbf{M}^j$:

$$T_n^j = \mu\left\{S_n(F, \mathbf{M}^j)\right\} + \sigma\left\{S_n(F, \mathbf{M}^j)\right\}.$$

Здесь $F$ — множество потоков длиной не меньше $n$. Данный алгоритм может быть использован для классификации в реальном времени.

Для проверки алгоритмов использовались данные, полученные в точке выхода из локальной сети в глобальную сеть. Проведена оценка правильности обоих алгоритмов классификации при исследовании TCP-потоков от клиента к серверу для протоколов HTTP, POP3 и SMTP. Правильность составляет более 90 %. Приведены параметры окна гауссова фильтра для обоих алгоритмов, которые были выбраны для достижения высокой эффективности. Оба алгоритма зависят от корректности процесса обучения (правильной идентификации потоков при обучении), причем показано, что второй алгоритм существенно более зависим, чем первый. Определенная трудность состоит в том, что в процессе работы классификатор должен зафиксировать начало каждого потока и идентифицировать его направление клиент — сервер, тем более что потоки от клиента к серверу оказались более информативными. Остается невыясненным, что произойдет в случае утери или переупорядочивания пакетов потока.

Близкие характеристики потоков рассматриваются в работе [6]. В частности, рассматривается возможность идентификации P2P-приложений на основе изучения распределения длин пакетов и интервалов между пакетами, относящимися к одному потоку (но без учета порядка прибытия). При этом поток предлагается делить на две части: обмен сигнальной информацией и обмен данными, так как сигнальное поведение существенно отличается от передачи данных. Предлагается использовать методы вейвлет-анализа (wavelet analysis [7]) для выявления различия между рассматриваемыми распределениями для P2P-приложений и остальных приложений. Приводятся результаты сравнения вейвлет-преобразований функции распределения межпакетного времени в сигнальных потоках и потоках данных для протоколов FTP и eDonkey. Выясняется, что на некоторых уровнях разложения вейвлет-коэффициенты существенно различаются. На основе этого различия можно провести классификацию. Работа [6] относится к числу методологических.

**3. Классификация IP-трафика с использованием контролируемого классификатора naive Bayes.** В работе [8] представлена техника разделения трафика на категории с применением контролируемого классификатора naive Bayes (NB). Ставилась задача на основе исследования тренировочных данных соотнести потоки тестового трафика с выбранными категориями. При этом каждый поток должен отображаться только на одну категорию, но сами категории необязательно должны быть уникальными в смысле приложения. Приводятся результаты деления на следующие категории: WWW, MAIL (протоколы imap, pop2/3, smtp), BULK (большие объемы, например ftp), SERV (сервисы, протоколы X11, dns, ident, ldap, ntp), DB (протоколы postgres, sqlnet oracle, ingres), P2P (KaZaA, BitTorrent, GnuTella), ATT (атаки, вирусы и черви), MMEDIA (мультимедиа, Windows Media Player, Real). Категории выбраны согласно образцам исследуемого трафика. Показано, что с использованием рассмотренной техники возможно разделение даже на группы минимально соотносящихся типов трафика. Представлена технология kernel estimation, позволяющая улучшить NB-метод и достичь большей эффективности. Кроме того, применен перспективный метод выбора атрибутов и сокращения избыточности fast correlation-based filter (FCBF), представленный в [9].

Для чистоты эксперимента тренировочные и тестовые данные состояли только из потоков TCP-трафика, причем потоки представляли собой семантически полные TCP-сессии (про-

слеживаются установление и разрыв TCP-соединения). Потоки формировались на основе полного содержания пакетов, захваченных в точке выхода из локальной сети в глобальную сеть. В качестве атрибутов потока первоначально выбраны длительность потока; TCP-порт; интервалы между пакетами (среднее, отклонение и др.); полезная нагрузка (среднее, отклонение и др.); эффективная пропускная способность, основанная на понятии энтропии [10]; преобразование Фурье функции плотности распределения интервалов между пакетами. Систематизированы свойства TCP-потоков, которые можно использовать для классификации трафика, в виде списка, включающего 249 атрибутов, представленных в [11].

NB-классификатор — это вероятностный классификатор, основанный на теореме Байеса. Термин "наивный" относится к предположению о независимости наблюдаемых переменных, от которых зависит принадлежность классу. Итак, рассматривается фиксированное множество классов $C = \{c_1, \ldots, c_k\}$. Требуется оценить величину $p(c_j|y)$ — вероятность того, что поток $y$, определяемый атрибутами, принадлежит классу $c_j$. С использованием формулы Байеса для условной вероятности получаем

$$p(c_j|y) = \frac{p(c_j)f(y|c_j)}{\sum\limits_{i=1}^{k} p(c_i)f(y|c_j)},$$

где $p(c_j)$ — вероятность получения класса $c_j$ независимо от наблюдаемых данных; $f(y|c_j)$ — функция плотности распределения (в дискретном случае — вероятность $y$ при заданном $c_j$); знаменатель — нормализующая константа. Таким образом, задача сводится к оценке распределения объектов для классов $p(c_j)$ и распределений $f(y|c_j)$ на основе тренировочных данных. Заметим, что вводятся два предположения: атрибуты независимы и имеют нормальные распределения внутри классов, параметры которых и надо оценить. В общем случае эти предположения неверны. Например, для класса WWW реальное распределение для атрибутов является мультимодальным, т. е. представляется смесью распределений. Несмотря на эти ограничения, классификатор обеспечивает получение более точных результатов по сравнению с более сложными методами, используемыми Data Mining.

В качестве примера вычислим $p(c_j|y)$ для случая, когда количество атрибутов равно единице (например, атрибут $a_1$), а количество классов — двум. Для тренировочных данных известно (определяется методом ручной проверки), сколько единиц данных принадлежит каждому классу, поэтому определяем $p(c_j) = n_{cj}/n$, где $n_{cj}$ — количество единиц, попавших в класс $c_j$; $n$ — общее число рассматриваемых единиц. Нормальное распределение характеризуется математическим ожиданием $\mu$ и дисперсией $\delta^2$, которые оцениваются через максимальное правдоподобие. Например, $\mu_1$ оценивается как отношение суммы значений атрибута $a_1$ для всех единиц класса $c_1$ к количеству единиц, принадлежащих классу, $\delta_1^2$ — как отношение суммы квадратов отклонений значений атрибута $a_1$ от среднего $(a_1 - \mu_1)^2$ по всем единицам данных к $n_{c1} - 1$. Зная характеристики распределения для класса и формулу функции плотности нормального распределения, можно вычислить $p(c_j|y)$.

Рассмотрим два классификатора: NB-классификатор и kernel estimator. Алгоритмически они совпадают. При использовании классического NB-метода параметры аппроксимируются нормальным распределением. Классификатор kernel estimator использует иной метод оценки параметров. Ядерная оценка (kernel density estimation [12]) — это способ непараметрической оценки функции плотности распределения случайной переменной на основе образцов значений. При наличии образцов $x_1, \ldots, x_n$ с неизвестной функцией распределения $f$ ядерная оценка $\hat{f}$ определяется по формуле

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

где $h$ — параметр сглаживания, называемый шириной (bandwidth); $K$ — ядро. Ядро — это любая функция, такая что

$$\int_{-\infty}^{+\infty} K(t)dt = 1.$$

В данном случае методика используется для оценки функции плотности вероятности $f(.|c_j), j = 1, \ldots, k$. Тогда

$$\hat{f}(t|c_j) = \frac{1}{n_{cj}h} \sum_{x_i : C(x_i) = c_j}^{n} K\left(\frac{t - x_i}{h}\right).$$

В качестве $K$ используется функция плотности нормального распределения с $\mu = 0,\ \sigma = 1$:

$$K(t) = \frac{1}{2\pi} \exp\left(\frac{-t^2}{2}\right).$$

В [13] показано, что по сравнению с NB-методом ядерная оценка обеспечивает более полную классификацию, если предположение о нормальности распределения неверное, и такую же, если оно является нормальным. Отмечается, что выбор параметра $h$ влияет на аппроксимацию, но он был выбран по умолчанию, принятому в реализации (www.cs.waikato.ac.nz/ml/weka/).

Еще одним средством, улучшающим характеристики классификации, является правильный выбор атрибутов. Во-первых, не должно быть неважных (irrelevant) атрибутов, которые не содержат информацию о классе. Во-вторых, не должно быть избыточных (redundant) атрибутов, сильно взаимосвязанных с другими атрибутами. Атрибут считается "правильным", если он важен для набора классов и не избыточен относительно других атрибутов. Для выбора "правильного" множества атрибутов технологии машинного обучения предусмотрены специальные методики, такие как фильтры (filters) и врапперы (wrappers) (см. [14]). В данном случае для нахождения "правильных" атрибутов до начала процесса классификации применяется фильтр FCBF.

При тестировании предложенного метода классификации использовались реализации (http://www.cs.waikato.ac.nz/ml/weka/) алгоритмов naïve Bayes, kernel density estimation и FCBF. Все данные предварительно были изучены вручную, чтобы получить достоверные результаты. Описание данных и процесс их предварительной идентификации приведены в [15]. Рассматриваются две меры эффективности: правильность относительно потоков и байтов и доверие. Сначала сравнивались NB-метод и kernel estimator на всем множестве атрибутов. В среднем NB-метод продемонстрировал правильность, приблизительно равную 65 %. При этом наибольшее доверие соответствовало классу MAIL, наименьшее — классам P2P (≈5 %) и ATT (≈1 %). Для kernel estimator средняя правильность приближенно равна 93 %, доверие к P2P повысилось до 23 %, а к ATT — до ≈9 %.

Далее выбирались "правильные" атрибуты. Для методик NB и kernel estimator количество атрибутов получилось разным. Наилучшие результаты у техники kernel estimator после фильтрации с помощью FCBF: правильность — ≈96 %, доверие возросло для всех классов:

для P2P — ≈36 %, для ATT — ≈13 %. Приводится список наиболее "правильных" атрибутов. Методика NB + FCBF проверялась на трафике, собранном годом позже, чем проводилось обучение. Оценка правильности приблизительно совпала, а доверие к некоторым классам возросло. Например, для P2P оно составило приблизительно 54 %. Таким образом, классификация новых данных может проводиться с использованием прежней модели.

Таким образом, установлено, что алгоритм чувствителен к первоначальному выбору множества атрибутов и функции их распределения. Показана необходимость проверки правильности выбора атрибутов для решения задачи классификации и приведен список "правильных" атрибутов.

**4. Идентификация приложений на основе изучения распределения размеров пакетов для TCP-соединений и взаимосвязи портов.** В работе [16] предложен подход к классификации трафика на основе изучения распределения размеров пакетов (packet size distribution (PSD)) TCP-соединения. Для каждого рассматриваемого соединения отбрасываются наиболее короткие управляющие пакеты (например, с флагами ACK/SYN) и пакеты с полной полезной нагрузкой (по размеру MTU). Пусть $SEQ_p =< p_{s1}, p_{s2}, \ldots >$ — последовательность длин пакетов, упорядоченная по убыванию пропорции (обозначим *pro*) появления пакетов с такими длинами: $pro(p_{si-1}) \geq pro(p_{si})$. Из последовательности $SEQ_p$ выбирается подпоследовательность $t$ элементов $DS =< p_{s1}, p_{s2}, \ldots, p_{st} >$, так чтобы сумма пропорций была больше некоторого порога, в данном случае 90 %. Последовательность $DS$ — это последовательность доминирующих размеров. Пропорция доминирующих размеров $DSP$ определяется по формуле $DSP =< pro(p_{s1}), pro(p_{s2}), \ldots, pro(p_{st}) >$. Рассматривается разность последующих размеров $p_{si} - p_{si-1}$. Среднее значение этой разности обозначим $CC$. Кроме того, для ускорения процесса распознавания вводится понятие таблицы port association table (PAT), элементами которой являются тройки <*src_ip*, *src_port*, *App*> и <*dst_ip*, *dst_port*, *App*>, если соединение <*src_ip*, *src_port*, *dst_ip*, *dst_port*> распознано как *App*. Кроме того, если один из адресов уже принадлежит PAT, а пара <*src_ip*, *src_port*> или <*dst_ip*, *dst_port*> имеет порт, номер которого на единицу отличается от записи в PAT, то считается, что данное соединение принадлежит той же сессии. Каждое соединение определяется как точка <*DS*, *DSP*, *CC*> в $(2n + 1)$-мерном пространстве, где $n$ — максимальное количество различных длин пакетов.

Сначала с использованием тренировочных данных исследуются приложения. Трафик каждого приложения порождался специально и по одному приложению, трафик остальных приложений отфильтровывался. Испытание проводилось по различным сценариям: при формировании потоков выбирались различные временные интервалы — 30 с, 5 мин и вся сессия; различная степень загрузки хостов, принимающих участие в испытаниях; различные источники файлообменного трафика; различные конфигурации приложений (пропускная способность, качество обслуживания (QoS) и протоколы с шифрованием и без шифрования). В качестве исследуемых приложений выбраны Apache, zFTP server, ShoutCast, представляющие потоковое приложение; WoW, представляющее игровое приложение; BitTorrent, eMule, Skype и MSN, представляющие P2P. Установлено, что разные приложения имеют разные PSD, а одинаковые типы приложений имеют одинаковые PSD.

Во время фазы обучения на тренировочных данных были вычислены центры приложений. Центр приложения определяется на основе информации о векторах всех соединений, соответствующих приложению. Предположим, что приложение A представлено $k$ векторами. Тогда центр определяется по формуле $C = \sum \mathbf{v}_i/k$, где $1 \leq i \leq k$, а $\mathbf{v}_i$ — $i$-й вектор. Заметим, что это значение может зависеть от реализации приложения.

Трафик, исследуемый классификатором, состоял из трафика приведенных выше приложений и представлял собой обмен данными между заранее известными хостами в реальной среде. Классификация включает два этапа. На первом этапе выделяются соединения, которые затем преобразуются в векторы вида $<DS,\ DSP,\ CC>$. На втором этапе проводится классификация соединений. При рассмотрении нового соединения сначала проверяется, удовлетворяют ли элементы соединения $<src\_ip,\ src\_port>$ или $<dst\_ip,\ dst\_port>$ правилам помещения в PAT. Если удовлетворяют, то пары помещаются в PAT с соответствующей пометкой о приложении. Если пары не помещаются в PAT, то для вектора соединения вычисляется евклидово расстояние от всех центров приложений. Евклидово расстояние между двумя соединениями $A$ и $B$ определяется по формуле

$$E\_Dist = \sqrt{\operatorname{dist}(DS_A, DS_B) + \operatorname{dist}(DSP_A, DSP_B) + (CC_A - CC_B)^2}.$$

где $\operatorname{dist}(x, y) = \sum\limits_{i=1}^{n} (x_i - y_i)^2$.

Если евклидово расстояние до центров всех приложений больше некоторого порогового значения, то приложение не может быть идентифицировано. Иначе соединение идентифицируется по наименьшему расстоянию от известного центра.

Проведенные исследования показали, что соединения, относящиеся к P2P, Apache и zFTP, корректно определяются в среднем в 98 % случаев, в то время как соединения для Skype-voice и MSN-voice chatting корректно определяются в 74 и 80 % случаев соответственно. При этом использование PAT для BitTorrent, eMule и zFTP ускорило распознавание, а для остальных приложений не повлияло на скорость распознавания. В среднем правильность составляет 96 %, а FP и FN — $4 \div 5$ %.

В данном случае предложен комбинированный подход — контролируемое обучение с соотнесением потоков на основе номеров портов. Контролируемое обучение имеет ряд недостатков. Вычисление центра приложения зависит от таких параметров, как тип коммуникационной линии, реализация приложения, состав тренировочных данных. Не исключено, что при классификации реального трафика вектор приложения, отличного от рассмотренного, будет близок к центру одного из рассмотренных приложений, тем более что в других работах распределение длин пакетов рассматривается наряду с другими атрибутами.

**5. Идентификация P2P-трафика на основе профилей приложений.** В работе [17] предложен двухступенчатый метод идентификации P2P-приложений. Отмечено, что P2P-приложения выявить труднее по сравнению с другими приложениями вследствие сложности их поведения. Поэтому предполагается сначала изучить поведение каждого приложения в отдельности, а затем среди всех потоков выделить те, которые имеют характерные признаки приложения определенного типа. Под потоком в данной работе понимается двунаправленный трафик между двумя хостами, определяемыми IP-адресами. Ниже указаны особенности предлагаемого подхода:

1) составление профилей приложения (профиль определяет коммуникационные шаблоны хоста, принимающего участие в обмене, соответствующем рассматриваемому приложению);

2) идентификация приложения на двух уровнях: сначала проверяется, принимает ли хост участие в обмене, соответствующем приложению, путем сравнения его профиля с уже изученными профилями, затем отделяются потоки, соответствующие приложению, от других потоков данных, относящихся к хосту;

3) выявление взаимосвязей потоков, в результате чего потоки, не имеющие отличительных признаков, могут быть соотнесены с уже помеченными потоками.

Профили приложений строятся с помощью контролируемого обучения с использованием алгоритма построения ассоциативных правил Apriori [18].

Ассоциативные правила строились для каждого приложения отдельно. Каждый набор тренировочных данных содержит трафик только одного P2P-приложения и трафики других приложений в качестве фона. В данной работе приведены результаты исследования двух P2P-приложений: BitTorrent и PPLive. Пометка тренировочных данных проводилась в основном с использованием IDS Bro [19].

При выборе атрибутов потоков использовались данные работы [8]. В качестве основных (axis) атрибутов потоков рассматриваются (отдельно для каждого направления) количество пакетов, количество байтов, размер первого пакета данных (для TCP — после процедуры "обмена рукопожатиями", для UDP — действительно первый), размер второго пакета (для TCP — после процедуры "обмена рукопожатиями"). Пятерки $<src\_ip$, $src\_port$, $dst\_ip$, $dst\_port$, $protocol>$ также относятся к разряду основных атрибутов потока. В качестве дополнительных (extra) атрибутов рассматриваются средний размер пакетов, отклонение от среднего размера пакетов, среднее для интервалов между пакетами, отклонение от среднего интервала между пакетами, общая длительность потока. Основные атрибуты важны для всех потоков, а дополнительные, скорее, для больших потоков. В число атрибутов входит также атрибут, значением которого является указатель приложения.

Так как алгоритм построения ассоциативных правил Apriori предполагает, что все атрибуты имеют булев тип, потребовалось преобразование атрибутов. Для преобразования непрерывных числовых атрибутов применялась методика разделения множества значений на "бункеры" (bins). В данном случае использовалось деление "equal-frequency", когда значения представляются в виде диапазонов, соответствующих часто встречающимся значениям. Для основных атрибутов выбиралось большое количество бункеров (20), для дополнительных — небольшое (5), так как, например, интервал между пакетами существенно зависит от состояния сети. Длительность потока вообще оценивалась как длинный поток и (или) короткий поток.

Использовалась реализация Apriori (www.borgelt.net//apriori.html). Реализация предлагает ряд возможностей, которые позволяют уменьшить число правил. Порождались только ассоциативные правила $A \Rightarrow B$, где $B$ — атрибут приложения.

На этапе составления профиля приложения порождались только максимальные (maximal) правила. Иными словами, если имеется два правила

$$r1 : A1 \Rightarrow B, \quad r2 : A2 \Rightarrow B,$$

такие что $A1 \subset A2$ и $r2$ удовлетворяет требованиям минимальной поддержки и доверия, то остается только $r2$. Кроме того, для множества правил выполнялся ряд преобразований. Правила представлялись в обобщенной форме. Например, правила, различающиеся только адресом и портом источника, представлялись как одно правило без указания конкретных адресов и портов; рассматривались только правила, содержащие не менее трех из основных атрибутов и т. д. Чтобы улучшить результаты построения профилей, алгоритм Apriori применялся к нескольким тренировочным наборам. Далее следовал этап слияния правил.

Как сказано выше, идентификация приложения проводится в два этапа. При одноэтапном подходе к классификации рассматривается поток, и если его атрибуты совпадают с одним из правил, то он помечается соответствующим приложением. В этом случае возникают две проблемы: 1) большая доля *FP*, так как не все шаблоны уникальны для приложения; 2) выявление только наиболее значимых шаблонов, что приводит к увеличению доли *FN*.

В предлагаемом подходе сначала все потоки группируются относительно хостов (источника либо получателя), например в зависимости от того, который из них является более активным, в рассматриваемом случае, относительно локального хоста. Хост помечается приложением, если его потоки удовлетворяют достаточно большому количеству правил и достаточно большое количество потоков удовлетворяет каждому правилу. После идентификации хостов начинается процесс идентификации потоков. При этом вместо максимальных правил рассматривается множество замкнутых (closed) правил (используются возможности реализации). Иными словами, если имеется два правила с одинаковыми значениями поддержки и доверия:

$$r1 : A1 \Rightarrow B, \quad r2 : A2 \Rightarrow B,$$

такие что $A1 \subset A2$, то исследуется соответствие потоков более узкому правилу $r1$. Кроме того, эти правила порождаются при меньших значениях параметров минимальных поддержки и доверия min_sup и min_conf, а для того, чтобы уменьшить число полученных правил, сохраняются только правила, соотносимые с первой группой правил, например, если новое правило и правило первой группы имеют в качестве значения атрибута *dst_ip* один и тот же адрес назначения. Следует отметить, что методика нахождения соотношений между потоками, удовлетворяющими разным правилам, не формализована и приведена в настоящей работе только в виде примеров.

При проведении эксперимента рассматривалось несколько наборов тренировочных данных (от четырех до восьми). Часть данных собиралась в точке выхода из локальной сети кампуса в глобальную сеть. Данные состояли из заголовков пакетов и 42 байт полезной нагрузки. Кроме того, трафик рассматриваемых приложений генерировался самостоятельно с использованием нескольких локальных компьютеров. Этот трафик идентифицируется с абсолютной правильностью. При исследовании BitTorrent каждый набор данных содержал 2000–5000 TCP-потоков и от 5000–50 000 UDP-потоков BitTorrent. При порождении правил с помощью Apriori значение параметра min_sup варьировалось в диапазоне от 100 до 300, а значение параметра min_conf полагалось равным 80 %. После соответствующих преобразований множество правил, используемых на первом этапе выявления деятельности хоста, содержало 15 и 12 правил, относящихся к TCP и UDP соответственно. Заметим, что UDP-деятельность более стабильна для разных наборов данных. Для выявления потоков PPLive рассматривалось 15 и 25 правил для TCP и UDP соответственно, для выявления потоков BitTorrent — 39 и 17 правил для TCP и UDP соответственно, для выявления потоков PPLive — 62 и 70 правил для TCP и UDP соответственно. На стадии тренировки основное время расходуется на обработку сырых данных, а также на выявление всех соединений и соединений, соответствующих рассматриваемым P2P-протоколам.

Для тестирования методики было рассмотрено 10 наборов данных, каждый из которых содержал трафик нескольких участников обмена, использующих исследуемые протоколы. Утверждается, что на уровне хоста правильность выявления равна 100 %, при этом для обоих приложений $FP = 0$. Для оценки алгоритма используется понятие средней правильности относительно рассматриваемых наборов проверочных данных. Для обоих рассматриваемых протоколов средняя правильность выявления TCP- и UDP-потоков превышает 90 %.

В настоящей работе не приводится информация о зависимости между выбранными атрибутами и результатами идентификации, однако отмечается, что такой параметр, как размер первого пакета данных, является достаточно значимым, поэтому в случае применения в P2P-протоколе методики использования произвольных размеров нескольких первых паке-

тов, соответствующих стадии договоренности между сторонами, качество выявления будет ниже. Проблемой является также выбор адекватных тренировочных данных.

Предлагаемая методика, как и многие другие, неприменима в случае шифрования на сетевом уровне, т. е. при недоступности информации заголовков пакетов. Применение методики в случае, если точкой обзора является не точка выхода из локальной зоны в Интернет, а опорная сеть, может быть существенно затруднено.

Ниже рассматривается задача порождения коммуникационных профилей без использования тренировочных данных или с использованием части тренировочных данных, а также задача построения обобщенных профилей, выявляющих "важное" поведение из неструктурированной массы сетевого трафика.

**6. Стабильность характеристик алгоритмов классификации относительно исследуемых данных и по истечении времени.** В работе [20] ставится задача сравнить алгоритмы классификации по скорости, эффективности и стабильности. Сравниваются четыре метода идентификации: идентификация путем изучения номеров портов (port-based); идентификация на основе исследования содержимого пакетов; идентификации с помощью классификаторов naïve Bayes и C4.5 (http://www.cs.waikato.ac.nz/ml/weka/). Два первых метода классификации относятся к числу детерминированных, два других, naïve Bayes и C4.5, являются вероятностными методами. В результате применения алгоритма единица данных получает метку наиболее вероятного класса. Выбор алгоритма C4.5 во многом обусловлен выводами, сделанными в работе [21], в которой этот алгоритм сравнивается с несколькими байесовскими методами, в том числе с байесовской сетью и байесовскими деревьями решений. Алгоритм C4.5 является более точным и быстрым (заметим, что множество атрибутов отличается от атрибутов, используемых в данной работе). В работе [22], в которой изучались алгоритмы построения деревьев решений, сделан вывод, что и в скорости обучения алгоритм C4.5 имеет преимущества по сравнению с другими алгоритмами. Недостатком этого алгоритма является построение деревьев больших размеров.

Изучается TCP/UDP-трафик в двух точках сети за несколько лет с целью проверки стабильности результатов классификации. Ставится задача выявить суть различных типов приложений и разделить трафик на категории согласно функциональности, а не по отдельным приложениям, тем более не по отдельным реализациям. При этом, если это P2P-трафик, использующий протокол HTTP, то он должен быть помечен как P2P.

Рассматриваются такие категории трафика, как Web-browsing, MAIL, BULK, CHAT, P2P, VoIP и др. Полный список категорий и распределение приложений согласно категориям приведены в работе [23].

Идентификация проводится на уровне потоков. В данном случае под потоком понимается двунаправленный трафик между хостами, определяемый пятеркой $<src\_ip$, $src\_port$, $dst\_ip$, $dst\_port$, $protocol>$. Для TCP (в случае, если не определено начало сессии) и UDP применяется механизм тайм-аута. В TCP-потоках выделяются направления клиент — сервер.

Предлагаемая методика имеет следующие особенности: 1) рассматриваются только данные, для которых установлена "абсолютная" истина; 2) обучение проводится с использованием небольшого набора "правильных" атрибутов, выделенных из большого набора эффективным методом; 3) имеется возможность исследовать только начало потоков, т. е. проводить классификацию в реальном времени; 4) используются уникальные способы определения стабильности методов классификации; 5) исследуется возможность классификации потоков с учетом состояния сессии (TCP) и без его учета (UDP); 6) проводится сравнение вычислительной сложности алгоритмов классификации.

Из трафиков разных лет в одной из точек сбора случайным образом были выбраны три дня: Day1, Day2, Day3 (за 2004, 2005, 2006 гг.), а в каждом из дней — по 10 30-минутных периодов. В другой точке сбора выделен один 30-минутный период SiteB (2007 г.). Из TCP-трафика были исключены неполные сессии, так как в основном это различные варианты сканирования. Среди таких потоков встречаются FTP-потоки, но обычно они содержат небольшое количество данных. Для TCP- и UDP-трафиков обучение и тестирование проводятся отдельно. Ниже рассматривается TCP-трафик (за исключением случаев, оговоренных особо).

При анализе с помощью методов naïve Bayes (с использованием kernel estimation и фильтра FCBF) и C4.5 рассматриваются следующие атрибуты потоков: счетчик пакетов с флагом PUSH (отдельно для направлений сервер — клиент, клиент — сервер); общее количество байтов, посланных в рамках TCP-окна [24], установленного при инициации соединения (отдельно для направлений сервер — клиент и клиент — сервер); средний размер сегмента (отношение количества байтов данных к количеству пакетов отдельно для направлений сервер — клиент и клиент — сервер); медиана от общего количества байтов в IP-пакетах; количество пакетов с полезной нагрузкой в направлении клиент — сервер; стандартное отклонение общего количества байтов в пакетах в направлении сервер — клиент; минимальный среди наблюдаемых размер сегмента в направлении клиент — сервер; порт сервера; порт клиента. Это подмножество стабильно относительно времени и точки обзора. Номера портов сервера и клиента входят в множество атрибутов, но не для соотнесения с приложениями согласно [25], а для соотнесения потоков с "абсолютной" истиной. Среди атрибутов выделяются две группы, содержащие по два атрибута, которые также хорошо отделяют сервисы друг от друга. Первая группа — стандартное отклонение общего количества байтов в пакетах в направлении клиент — сервер от общего количества байтов, посланных в рамках инициализационного TCP-окна в направлении клиент — сервер. Вторая группа — счетчик пакетов с флагом PUSH в направлении сервер — клиент по отношению к минимальному из наблюдаемых размеров сегмента в направлении сервер — клиент.

Вычисление значения одного атрибута для потока длиной $n$ требует выполнения порядка $O(n \log_2 n)$ операций (например, для вычисления медианы длин пакетов) и затрат памяти на уровне $O(n)$. Соответственно $k$ потоков будут обработаны за время, не превышающее $O(kn \log_2 n)$.

Методика определения "абсолютной" истины представлена в [23]. Каждый поток тщательно исследовался с применением методов изучения полезной нагрузки пакетов, известных номеров портов, известных адресов и имен хостов, использования дополнительной информации о некоторых хостах, пользователях, поведении приложений. Заключение делалось на основе выводов, полученных с помощью нескольких методов, чтобы исключить ошибки.

При тестировании метода идентификации на основе номеров портов рассматривались известные номера портов, меньшие или равные 1023, зафиксированные в [25]. Номер порта определялся по первому пакету потока в направлении клиент — сервер. В рассматриваемом трафике обнаружено 35 таких портов, из них только 16 соответствовали ассоциированному приложению.

При тестировании метода глубокого исследования пакетов потоки проверялись на наличие соответствующих сигнатур или соблюдение известной семантики протоколов. Как показано в [26], некоторые потоки могут быть идентифицированы таким способом по первому пакету потока. Однако иногда прежде чем сделать заключение, нужно исследовать до 1 кбайт полезных данных. Использован механизм, применяемый в [27]. Поток помечается приложением, если найдена соответствующая сигнатура хотя бы в одном из направлений

потока. Поиск осуществляется в первых 10 пакетах потока. Далее этот метод будем называть L7.

При тестировании алгоритмов naïve Bayes и C4.5 классификация проводилась на основе изучения атрибутов первых пяти пакетов. В данной работе приводится обоснование рассмотрения пяти пакетов. Для этого изучалась правильность классификации для разных окон обзора. Окно — это либо заданное количество пакетов $n$, либо меньшее количество в случае тайм-аута (в данном случае 15 с). Рассматривались окна с количеством пакетов от 4 до 10. Установлено, что наиболее высокая аккуратность достигается при $n = 5$ и она сравнима с аккуратностью, достигаемой при изучении полных TCP-потоков.

Ниже приводятся результаты сравнения применяемых методик между собой и каждой методики относительно времени сбора данных, а именно проводились тесты: T1 — эффективность алгоритмов (правильность, точность, полнота) для каждого блока данных; T2 — для C4.5 точность, полнота относительно категорий трафика для каждого блока данных; T3 — деградация правильности алгоритмов со временем: обучение на блоках Day1, Day1 + Day2 (2003–2004 гг.), проверка на блоках Day2, Day3 (2006 г.); T4 — деградация правильности C4.5 относительно категорий трафика со временем: обучение на блоках Day1, Day1 + Day2, проверка на блоках Day2, Day3; T5 — сравнение алгоритмов в условиях обучения на одном блоке данных и применения к другому блоку данных, исследуются Day3 и SiteB, собранные в разных точках сети и в разное время; для C4.5 — дополнительно сравнение относительно классов; T6 — сравнение скорости обучения и тестирования для C4.5, naïve Bayes и L7.

При тестировании T1 правильность C4.5 относительно потоков, пакетов и байтов составляла более 99 % для всех блоков данных. Naïve Bayes имеет больший разброс правильности относительно блоков данных, однако для потоков правильность составила не менее 96 %, для байтов и пакетов — не менее 80 %. Хуже всего характеристики у L7 даже с набором правил от 2008 г., особенно в терминах байтов.

При тестировании T2 всех блоков данных на точность и полноту относительно категорий трафика для C4.5 низкие показатели у категории ATTACK, это объясняется слишком большим разнообразием поведения приложений этой категории. Для P2P оба показателя не опускаются ниже 96 %. При тестировании T3 деградация правильности C4.5 оказалась равной 2 % за год, а для L7 и port-based — 5 %. При проверке деградации правильности относительно категорий трафика для C4.5 (тестирование T4) для многих категорий деградация оказалась небольшой, в том числе для P2P. Однако для некоторых категорий выявилась значительная деградация, особенно по показателю полноты. В тестировании T5 рассматривались блоки Day3 и SiteB. Различаются два этапа. На первом этапе обучение проводится на одном блоке данных, а проверка — на другом. На втором этапе обучение проводилось на блоке, состоящем из половины трафика Day3 и половины трафика SiteB, а тестирование — на каждой из вторых половин блоков Day3 и SiteB. На первом этапе вновь наилучшие результаты правильности дало применение C4.5. Относительно успешное применение L7 обусловлено существенной долей Web-трафика в 2007 г., для которого хорошо представлены сигнатуры. Для некоторых категорий, например для P2P, при использовании алгоритма C4.5 с обучением на блоке Day3 при проверке на блоке SiteB полнота снизилась до 58 %; во многом ухудшение результатов зависит от того, что трафик P2P мало представлен в блоке Day3 и широко представлен в SiteB. Некоторые категории плохо представлены в обоих блоках. Тестирование на втором этапе не выявило существенного различия в правильности для блоков Day3 и SiteB для всех методик. Для отдельных категорий выявлена

разница при тестировании C4.5. Это объясняется недостаточным количеством потоков категории, что не позволяет провести адекватное обучение. В тестировании T6 методики C4.5 и naïve Bayes сравнивались относительно скорости обучения на блоках разных размеров. Например, на 30-минутном блоке обучение naïve Bayes происходит в пять раз быстрее, чем обучение C4.5. Нормализованное (относительно количества потоков) время вычисления атрибутов и время классификации алгоритмом L7 представляет собой практически константу. Однако время вычисления атрибутов составляет почти 1/3 времени классификации. Время классификации методикой C4.5 достаточно небольшое и зависит от глубины "дерева". При увеличении размеров блока во время обучения глубина незначительно возрастает, при этом нормализованное время классификации практически равно константе. Для методики naïve Bayes нормализованное время классификации не является константой, сложность модели значительно увеличивается.

Для классификации UDP-трафика, так же как и для TCP, применялась методика окна, равного пяти, но таймер устанавливался равным 60 с. С помощью фильтра FCBF найдено следующее множество атрибутов: количество пакетов в обоих направлениях; минимум байтов полезной нагрузки в направлении клиент — сервер; минимум байтов полезной нагрузки в направлении сервер — клиент; максимум байтов полезной нагрузки в направлении клиент — сервер; максимум байтов полезной нагрузки в направлении сервер — клиент; количество байтов полезной нагрузки в направлении клиент — сервер, прежде, чем будет получен ответ; количество пакетов в направлении клиент — сервер, прежде чем будет получен ответ; порт сервера; порт клиента.

Для UDP-трафика также проводились сравнения показателей T1-T5. Правильность всех методик практически не изменилась, а для L7 даже повысилась. Некоторые категории не представлены в UDP-трафике, для представленных категорий показатели C4.5 не ниже 92 %. При сравнении показателей на блоках Day1 и Day3 для традиционных категорий практически не отмечается деградации, в отличие от P2P. При обучении на блоке Day3 и тестировании на блоке SiteB также отмечается деградация для P2P-трафика.

Показано, что совместное использование окна наблюдения и метода C4.5 позволяет классифицировать TCP- и UDP-трафики по категориям в реальном времени. Лучшие результаты показал алгоритм C4.5 при проверке изменения эффективности относительно времени и блоков данных, на которых проводится обучение.

**Заключение.** В работе рассмотрены различные постановки задачи классификации: классификация трафика согласно категориям, приложениям или выявление трафика категории P2P как наиболее трудно идентифицируемой. Объектами классификации являются потоки: однонаправленные, двунаправленные, полные TCP-сессии, IP-трафик между хостами. Для вычисления атрибутов, соответствующих потокам, используется информация о заголовках пакетов и времени их прибытия. "Абсолютная" истина о принадлежности потока к категории трафика строится на основе полного содержимого пакетов.

При исследовании алгоритмов изучены факты, влияющие на качество классификации: 1) важность выбора набора непротиворечивых и неизбыточных атрибутов и зависимость набора от образцов данных; 2) необходимость выбора различных наборов атрибутов и ассоциативных правил для TCP- и UDP-потоков, несмотря на то что оба типа потоков могут одновременно использоваться одним и тем же приложением, как, например, в случае P2P; 3) чувствительность к выбору предполагаемых параметров распределения значений атрибутов внутри класса; 4) применимость алгоритма к однонаправленным потокам трафика; 5) влияние размеров блоков образцов данных.

Рассмотрены особенности контролируемой и частично контролируемой классификации и алгоритмов анализа данных применительно к классификации трафика. Зависимость от реализации приложений можно ослабить, если чаще проводить переобучение. Однако в этом случае возникает проблема сложности проверки: для небольших объемов данных проще установить "абсолютную" истину относительно представленных классов приложений, но недостоверны параметры эффективности, а большие объемы трудно проанализировать.

Проведение сравнения результатов классификации, представленных в различных работах, существенно затруднено вследствие использования различных мер эффективности алгоритмов, вычисления эффективности относительно различных структур данных (байты, пакеты, потоки) и различия исследуемых блоков данных.

Сами алгоритмы имеют различия в вычислительной производительности, производительности процесса обучения, для деревьев решений — в размерах данных. Установлено, что деревья решений обеспечивают большую точность. Рассмотренный алгоритм C4.5 обладает также свойством стабильности относительно времени сбора и содержимого блоков данных, на которых проводится обучение.

В целом, несмотря на наличие различных ограничений и предположений (о независимости атрибутов, о нормальности распределения числовых атрибутов внутри класса, о правильности порядка прибытия пакетов, о минимальном влиянии сетевого окружения на задержки между пакетами), можно сделать вывод, что рассматриваемые методы являются достаточно эффективными. Наиболее сложным в распознавании является P2P-трафик.

## Список литературы

1. Барсегян А. А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, Olap / А. А. Барсегян, М. С. Куприянов, В. В. Степаненко, И. И. Холод. СПб.: БХВ-Петербург, 2007.

2. Mitchell T. Machine learning. N. Y.: McGraw-Hill, 1997.

3. Гмурман В. Е. Теория вероятностей и математическая статистика. М.: Юрайт, 2011.

4. Crotti M., Dusi M., Este A., et al. Application protocol fingerprinting for traffic classification // Annual GTTI award for PhD theses in the field of communication technologies, 2007. [Electron. resource]. http://www.gtti.it/GTTI07/papers/Dusi_unibs.pdf.

5. Haddad R. A., Akansu A. N. A class of fast Gaussian binomial filters for speech and image processing // IEEE Trans. Acoust., Speech Signal Proc. 1991. V. 39. P. 723–727.

6. Dedinski I., De Meer H., Han L., et al. Cross-layer peer-to-peer traffic identification and optimization based on active networking // Proc. of the 7th Intern. workshop on active networks (IWAN 2005), Sophia-Antipolis (France). Nov. 21–23, 2005. Berlin; Heidelberg: Springer-Verlag, 2009. P. 13–27.

7. Chui Ch. K. An introduction to the wavelets. N. Y.: Acad. Press, 1992.

8. Moore A. W., Zuev D. Internet traffic classification using Bayesian analysis techniques // ACM SIGMETRICS 2005, Banff, Alberta (Canada), June 2005. N. Y.: ACM, 2005. P. 50–60.

9. Yu L., Liu H. Feature selection for high-dimensional data: A fast correlation-based filter solution // Proc. of the 20th Intern. conf. on machine learning (ICML 2003), Washington, 2003. Palo Alto: AAAI Press, 2003. P. 856–863.

10. Duffield N. G., Lewis J. T., O'Connell N., et al. Entropy of ATM traffic streams // IEEE J. Select. Areas Commun. 1995. V. 13, iss. 6. P. 981–990.

11. Moore A. W., Zuev D. Discriminators for use in flow-based classification: Tech. report / Intel Res. Cambridge, 2005.

12. Wand M. P. Kernel smoothing / M. P. Wand, M. C. Jones. L.: Chapman and Hall/CRC, 1995.

13. John G., Langley P. Estimating continuous distributions in Bayesian classifiers // UAI'95: Proc. of the 11th conf. on uncertainty in artificial intelligence, Quebec (Canada), 1995. San Francisco: Morgan Kaufmann, 1995. P. 338–345.

14. Guyon I., Elisseeff A. An introduction to variable and feature selection // J. Machine Learn. Res. 2003. V. 3. P. 1157–1182.

15. Moore A. W. Discrete content-based classification — a data set: Tech. report / Intel Res. Cambridge, 2005.

16. Lin Y-D., Lu Ch-N., Lai Y-Ch., et al. Application classification using packet size distribution and port association // J. Network Computer Appl. 2009. V. 32. P. 1023–1030.

17. Hu Y., Chiu D- M., Lui J. C. S. Profiling and identification of P2P traffic // Comput. Networks. 2009. V. 53. P. 849–863.

18. Agrawal R., Srikant R. Fast algorithms for mining association rules // Proc. of the 20th VLDB conf., Santiago de Chile (Chile), Sept. 12–15, 1994. San Francisco: Morgan Kaufmann, 1994. P. 487–499.

19. Paxson V. Bro: A system for detecting network intruders in real-time // Comput. Networks. 1999. V. 31, N 23/24. P. 2435–2463.

20. Li W., Canini M., Moore A. W., Bolla R. Efficient application identification and the temporal and spatial stability of classification schema // Comput. Networks. 2009. V. 53, N 6. P. 790–809.

21. Williams N., Zander S., Armitage G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification // SIGCOMM Comput. Commun. Rev. 2006. V. 36, iss. 5. P. 5–16.

22. Lim T.-S., Loh W.-Y., Shih Y.-S. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms // Machine Learn. 2000. V. 40, iss. 3. P. 203–229.

23. Canini M., Li W., Moore A. W., Bolla R. GTVS: Boosting the collection of application traffic ground truth // Lecture Notes Comput. Sci. 2009. V. 5537. P. 54–63.

24. Transmission Control Protocol. [Electron. resource]. http://www.ietf.org/rfc/rfc793.txt.

25. Internet assigned numbers authority (IANA). [Electron. resource]. http://www.iana.org/assignements/port-numbes.

26. Moore A. W., Papagiannaki K. Toward the accurate identification of network applications // Lecture Notes Comput. Sci. 2005. V. 3431. P. 41–54.

27. Application layer packet classifier for Linux. [Electron. resource]. l7-filter.sourceforge.net.

*Щербакова Наталья Григорьевна — ст. науч. сотр. Института вычислительной математики и математической геофизики СО РАН; e-mail: nata@nsc.ru*

Дата поступления — 11.09.12