Prog Artif Intell (2012) 1:223–234
DOI 10.1007/s13748-012-0019-5

REGULAR PAPER

Two methods for reliable classification of network traffic

Mikhail Dashevskiy · Zhiyuan Luo

Received: 10 November 2011 / Accepted: 16 May 2012 / Published online: 22 June 2012

Abstract Accurate classification of network traffic can offer substantial benefits to service differentiation, enforcement of security policies and traffic engineering for network operators and service providers. Machine learning algorithms have been used to classify network traffic with good results. However, not knowing the confidence of these classifications makes it difficult to measure and control the risk of error using a decision rule. Modern network resource management systems are becoming increasingly complex and as such require high quality, reliable predictions with confidence measures. These reliability measures allow service provider and network carrier to effectively perform a cost-benefit evaluation of alternative actions and optimise network performance such as delay and information loss. In this paper, we consider the problem of reliable network traffic classification. Two recently developed machine learning methods, namely Conformal Predictor and Venn Probability Machine, are presented for application in network traffic classification. These two methods are based on the identically independently distributed sequence of data instances assumption. Experiments on publicly available real network traffic datasets in the on-line setting show these two methods can perform well and produce reliable classifications. Comparison is also made between these two methods.

Keywords Network traffic classification · Reliable classification · Conformal predictors · Venn probability machines · Validity · Efficiency

M. Dashevskiy · Z. Luo (✉)
Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK
e-mail: zhiyuan@cs.rhul.ac.uk

1 Introduction

The Internet is a globally distributed network of networks supporting many applications and services. The Internet is based on a packet switching technology where the exchanged messages are divided into packets before they are sent. Each packet has a header and payload where the header carries the information that will help the packet get to its destination such as the sender's Internet Protocol (IP) address. The payload carries the data in the protocols that the Internet uses. Typically a stream of packets is generated when a user visits a website or sends an email. A traffic flow is uniquely identified by 4-tuple {source IP address, source port number, destination IP address, destination port number}. Two widely used transport layer protocols are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The main differences between TCP and UDP are that TCP is connection oriented and a connection is established before the data can be exchanged. On the other hand, UDP is connectionless.

The Internet traffic is a huge mixture and there are thousands of different applications which generate lots of different traffic. In addition to "traditional" applications (e.g. Email), new Internet applications such as games and peer-to-peer (P2P) file sharing become popular. There are different packet sending and arrival patterns due to interaction between the sender and the receiver and data transmission behaviour. Traffic classification can be defined as methods of classifying traffic data based on features passively observed in the traffic, according to specific classification goals. Classification of network applications responsible for the generation of the traffic flows is important for network management tasks such as monitoring trends of the applications in operational networks and effective network planning and design. In particular, accurate and fast classification of network traffic based on statistical flow characteristics can offer substantial

benefits to a number of key areas in Quality of Service support and service differentiation, enforcement of security policies, traffic engineering and support for Service Level Agreements for many network operators and service providers [15]. The requirements for such classification of Internet traffic are increasing rapidly due to availability of high capacity communications links, heterogeneity of Internet traffic types and continued evolution of applications such as tunnelling and end-to-end encryption. Different approaches have been developed to traffic classification over years, for example port based [30], host behaviour-based "fingerprinting" [13,14] and signature-based approaches [27]. However, the port-based classification is ineffective for classifying P2P applications, 30–70 % of the Internet traffic is classified as "unknown" [21]. In additional, emerging applications and services avoid the use of well-known ports altogether. More recently, flow-based classification has received much attention [23,24]. This approach performs classification based on various flow features, such as the number and size distribution of packets in a flow, flow duration and inter-packet arrival time [9,19]. In this paper, we focus on the problem of network traffic classification using flow-level statistics.

The increase in the diversity of applications makes network tasks like administration, network planning and policing difficult. The classification of traffic is an enabling function for a variety of other applications and topics, including Quality of Service, security and intrusion–detection. Accurate and reliable application identification methods can make resource allocation more effective, since this variability in applications creates conflicting resource requirements that are difficult to fulfill. Multiclass traffic classification provides fine control on these applications. Accurate traffic classification requires algorithmic capability, in particular, machine learning algorithms. The idea of using machine learning techniques for Internet traffic classification is not new. Many machine learning algorithms have been successfully applied to the problem of network traffic classification [10,23,31]. Usually these algorithms provide predictions in the form of a single class label with no measure of how confident the algorithm is in that prediction. However, not knowing the confidence of predictions makes it difficult to measure and control the risk of error using a decision rule. Modern network resource management systems are becoming increasingly complex and as such require high quality, reliable predictions to optimise performance and satisfy user demands. Therefore, introducing more reliable algorithms for network resource management will result in higher quality of service for network users. In such circumstances, a measure of confidence is essential and a confidence measure would constitute important meta-knowledge about the learner [16]. Recently developed frameworks of conformal predictors and Venn probability machines [29] allow us to address these problems. Conformal predictors output region predictions with the guaranteed asymptotical error rate and Venn probability machines output multiprobability predictions which are valid in respect of observed frequencies. These frameworks are based on the i.i.d. (independent and identically distributed) sequence of data instances assumption. These frameworks have been successfully applied in a number of applications such as medical diagnosis, network traffic classification and estimation of effort for software projects [2,5,6,17,26].

Our previous work investigated these two methods and their possible applications [6]. This paper focuses on the application of these two frameworks to network traffic classification. Extensive experimental results on real traffic datasets are presented. A fully comprehensive discussion and comparison of these two approaches is given. The remainder of the paper is structured as follows. Related approaches to performance with guarantees in machine learning and their limitations are discussed in Sect. 2. Two methods for reliable classification, namely conformal predictors and Venn probability machines are presented in Sect. 3. Experimental results of both conformal predictors and Venn probability machines on publicly available real traffic datasets are shown in Sect. 4. Finally, some discussions and conclusions are given in Sect. 5.

2 Performance with guarantees in machine learning

All predictions suffer from some degree of uncertainty. Reliable estimation of confidence remains a significant challenge as machine learning algorithms proliferate into many challenging real world applications. In machine learning, we are typically given a set of training examples $z_1, z_2, \ldots, z_n$. Each example z_i consists of an object x_i and its label y_i, $z_i = (x_i, y_i)$. The objects are elements of an object space $\mathbf{X}$ and for the classification the labels are elements of a finite label space $\mathbf{Y}$. For example, in network traffic classification, x_i are multi-dimensional traffic flow statistics vectors and y_i are application traffic groups taking only finite number of values such as {WWW, EMAIL, P2P}. Machine learning algorithms take a training set $(z_1, z_2, \ldots, z_n)$, form models and make predictions about the future new examples. The goal of machine learning algorithms is to produce predictors having the smallest possible risk (expected loss) on new examples where loss occurs if a wrong prediction is made. For supervised classification problems, it is important that the behaviour of machine learning algorithms is well understood, that its results can be controlled and come with strict performance guarantees.

There are different approaches that allow users to assess total accuracy or hedge each prediction. Among them are such well known ones as statistical learning theory, Bayesian learning and confidence interval estimation. In statistical

learning theory [20,28], a single label prediction is produced for each object based on the training set and there is a theoretical guarantee that these predictions become more accurate with greater probability while the training set size increases. This means that the probability of error will not exceed a certain threshold ε unless an event of the preset probability δ has happened. We set the value of probability δ and calculate ε, which is an expected loss of the total population. However, these bounds of error ε may often be too loose to be informative [11,20]. Error bounds provided by statistical learning theory are not preset, but are computed. In addition, these error bounds give no measure of uncertainty for individual predictions.

By contrast, Bayesian learning [4] and other probabilistic algorithms may complement each individual prediction with such additional information. However, the main disadvantage of these algorithms is that they often depend on strong statistical assumptions used in the model. When the data conforms well with the statistical model, Bayesian learning outputs valid predictions. However, if the data does not match the model (or the a priori information is not correct), which is usually the case for real-world data, predictions may become invalid and misleading [12]. When applied to classification problems, most probabilistic algorithms output a singleton prediction with assigned probability, which is a label with the highest posterior probability. This implies that the probability of the output is not controlled. Some Bayesian methods assume the independence between features (for example, a naive Bayes classifier). But this is a further restriction on the model and it does not usually hold true for real-world data. It is possible to model the dependencies among the features, for example, using Bayesian belief networks. But a Bayesian belief network has to be learned or imposed. In this case, the solution of the problem gets computationally inefficient [22].

Confidence intervals (CI) are used in statistics as an interval estimate of a population parameter to indicate the reliability of an estimate: we obtain confidence intervals (instead of a single value) that estimate the parameter and a confidence level indicates how likely the value lies within the confidence interval. The calculation of a confidence interval generally requires certain assumptions about the statistical model. For example, it may be based on the assumption that the distribution of the sample population is normal. CI can be also applied in machine learning. In this case, we fix a simple predictor and consider the error rate of a predictor (i.e., the expectation of the loss of the simple predictor) as a parameter to estimate. We can then produce confidence intervals which estimate predictor's error relying on errors at different steps. This approach is called hold-out estimates [29] and is a technique for assessing how the results will be generalised to an independent data set. We can choose a confidence level and apply the method to compute a CI for the true performance. However, CIs output an estimate of the accuracy of the algorithm in general and do not supply an information regarding how reliable a prediction for each individual object is. Also CIs when used in hold-out estimates are based on the assumptions that the errors (rather than examples) are i.i.d. Therefore, this assumption does not take the nature of learning process into consideration, because in real learning, algorithms take into account the information on previously made errors. This approach has been criticised in [3].

In the next section, two recently developed frameworks of conformal predictors and Venn probability machines will be presented and discussed. These frameworks allow us to complement each individual predictions with its reliability measure and these predictions have performance guarantees in an on-line setting [29].

3 Two methods for reliable classification

A performance guarantee can be defined in different ways. It can be a guarantee that the number of wrong predictions divided by the total number of predictions (i.e. the error rate) of an algorithm converges to a pre-defined level with the number of predictions going to infinity. It can also be a guarantee that a set of probability distributions on the possible outcomes output by a prediction algorithm consists of distributions close to each other. This section presents two methods for reliable classification: conformal predictors and Venn probability machines. These newly developed methods have several advantages. Firstly, they hedge every prediction (providing additional information regarding how strongly we believe it) individually rather than estimate an error on all future examples as a whole. As a result, the supplementary information which is assigned to predictions and reflects their reliability is tailored not only to the previously seen examples (a training set) but also to the new object. Secondly, both conformal predictors and Venn probability machines produce valid (or well-calibrated) results. Validity is an important property and in this case has a form of a guarantee that the error rate of region predictions converges to or is bounded by a pre-defined level when the number of observed examples tends to infinity or that a set of output probability distributions on the possible outcomes agrees with observed frequencies. The property of validity is based on a simple i.i.d. assumption. More importantly, the property of validity is theoretically proved in the on-line mode [29].

3.1 Conformal predictors

Conformal predictors are based on the principles of algorithmic randomness (closely related to Kolmogorov complexity), transductive inference and hypothesis testing [29]. Conformal predictors make classifications according to how similar the current example is to the representatives of

different objects' classes. This is done based on the idea of *Nonconformity Measure*. To give an idea on what a nonconformity measure is, we need to use the term *bag*. A bag (a multiset) is a set of elements in which repetition is allowed. Given a nonconformity measure (A_n) and a bag of examples $\wr z_1, \ldots, z_n \int$ we can calculate the nonconformity score (real value): $\alpha_i = A_n(\wr z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n \int, z_i)$ for each example in the bag. The nonconformity measure indicates how dissimilar an example is from a group of examples.

Algorithm 1 Conformal predictor algorithm for classification

Require: training examples $\{z_1 = (x_1, y_1), z_2 = (x_2, y_2), \ldots, z_n = (x_n, y_n)\}$, label space $\mathbf{Y}$, new example x_{n+1}
Require: non-conformity measure A, confidence level γ
  for $y \in \mathbf{Y}$ **do**
    $z_{n+1} = (x_{n+1}, y)$
    for j in $1, 2, \ldots, n+1$ **do**
      $\alpha_j = A(\wr z_1, \ldots, z_n, z_{n+1} \int, z_j)$
    end for
    $p(y) = \frac{\#\{j=1,\ldots,n+1, \alpha_j \geq \alpha_{n+1}\}}{n+1}$
  end for
  Output: predictive set $R^{\gamma}_{n+1} = \{y : p(y) \geq 1 - \gamma\}$
  Output: single prediction $\hat{y}_{n+1} = \arg\max_y \{p(y)\}$
  Output: confidence $conf(\hat{y}_{n+1}) = 1 - \max_{y \neq \hat{y}_{n+1}} \{p(y)\}$
  Output: credibility $cred(\hat{y}_{n+1}) = \arg\max_y \{p(y)\}$

Now, to use the nonconformity measure we want to compare the current example (which we are trying to classify) and the examples from the training set. In this way, we can see how the current example fits into the "whole picture" of the dataset. The conformal predictor calculates $p(y)$ defined as [2,29]: $p(y) = \frac{|\{j=1,\ldots,n+1 : \alpha_j \geq \alpha_{n+1}\}|}{n+1}$ which is a *p-value* associated with a hypothetical completion $y_{n+1} = y$ for the test example x_{n+1} (see Algorithm 1). Using these *p-values*, predictions made by conformal predictors can be presented in the following two ways:

- Either the user inputs a required degree (level) of certainty and the algorithm outputs a predictive set: a list of classifications that meet this confidence requirement.
- Or the algorithm outputs a so called *single prediction*: an individual predictions together with its confidence, which is the minimal level of certainty at which the output is certain (predictive set consists of only one label) and credibility, which indicates whether the test object is representative of the training set, in order to compare with the other conventional learning methods.

The common way of presenting the prediction results is to preset a significance level ("degree of certainty") $\gamma < 1$ and output the (γ)*-predictive set (region)* containing all labels with p-value equal or greater than $1 - \gamma$:

$$R^{\gamma}_{n+1} = \{y : p(y) \geq 1 - \gamma\} \quad (1)$$

Prediction errors can occur when the prediction set fails to contain the true label. If the size of the predictive set R^{γ}_{n+1} is 1 or 0, then this prediction is *certain*, otherwise it is uncertain and we have multiple predictions. In the context of conformal predictors, uncertain predictions are not errors, but these are indications that the amount of information is not sufficient to make a certain decision at the selected level. Naturally, the higher the confidence level is, the more multiple predictions will appear.

The main advantage of conformal predictors is their property of validity: the asymptotic number of errors, that is, erroneous region predictions, can be controlled by the significance level—the error rate we are ready to tolerate which is predefined by the user [29]. However, the property of validity is achieved at the cost of producing region predictions: instead of outputting a single label as a prediction, we may produce several of them any of which may be correct. Such multiple predictions are not mistakes: they are output when the conformal predictor is not provided with sufficient information for producing valid predictions at a certain error rate. Informativeness, or in other words efficiency, of a conformal predictor can be translated as its ability to produce as small region predictions as possible. Thus, we have to balance validity (the error rate) and efficiency (the number of labels in each prediction): lower error rates will result in larger region predictions, and vice versa. This feature makes conformal predictors a very flexible tool.

3.2 Venn probability machine

Machine learning applications may require prediction of a label complemented with probability that this prediction is correct. Venn probability machine is a multiprobability classification system. The term multiprobability means that we announce several probability distributions for the new label rather than a single one. Venn probability machines are based on the idea of dividing examples into groups and, when a new object arrives, somehow assigning it to one of the groups. Then we can use the frequencies of labels in the group containing the current object as probabilities for the new object's label [5,29].

More formally, let n be a predefined number of examples used as the input by the Venn probability machine. A *Venn taxonomy* is a sequence A_k, $k = 1, \ldots, n$, where each A_k is a finite partition of the space $\mathbf{Z}^{(k-1)} \times \mathbf{Z}$. The expression $\mathbf{Z}^{(k-1)} \times \mathbf{Z}$ here means that A_k does not depend on the order of the first $k-1$ arguments. Let $A_k(\omega)$ be the element of the partition (A_k) that contains $\omega \in \mathbf{Z}^{(k-1)} \times \mathbf{Z}$. With every taxonomy (A_k) we associate a Venn probability machine. After choosing a taxonomy, we can start making predictions. Each prediction is a class associated with the current example by the Venn probability machine.

Let us call the class of possible examples $\mathbf{Y}$. For each new example x_n, the classifying system has to assign the new example to one of the classes $\mathbf{Y}$. Firstly, we consider a class $y \in \mathbf{Y}$ and partition $\{z_1 = (x_1, y_1), \ldots, z_n = (x_n, y)\}$ into categories, assigning two points in $\mathbf{Z}$ to the same category if and only if

$$A_n(\wr z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n \int, z_i) = A_n(\wr z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_n \int, z_j)$$

$\wr z_1, \ldots, z_n \int$ means a bag.

The category T containing $z_n = (x_n, y)$ is always non-empty. Let p_y be the empirical probability distribution of the labels in this category T:

$$p_y\{y'\} := \frac{\left|\{(x^*, y^*) \in T : y^* = y'\}\right|}{|T|}, \quad y' \in \mathbf{Y}$$

This distribution depends implicitly on the object x_n and its hypothetical label y. Trying all possible hypotheses of the label y_n being equal to y, the Venn probability machine determined by the taxonomy generates a set of distributions $P_y := \{p_y : y \in \mathbf{Y}\}$ for all possible labels y. Thus as the output of Venn probability machine, we obtain as many distributions as the number of possible labels. However, this output can be interpreted in a simpler way. We can force Venn probability machines to make singleton predictions so that each prediction is complemented with an interval that the prediction is correct. This can be made as follows. After calculating empirical probability distributions P_y, we calculate the quality of each prediction $y' : q(y') = \min_{y \in \mathbf{Y}} P_y\{y'\}$ and then predict the label with the highest quality $y_{pred} = \arg\max_{y \in \mathbf{Y}} q(y')$. We can complement this singleton prediction with a probability interval $[\min_{y \in \mathbf{Y}} P_y(y_{\text{pred}}), \max_{y \in \mathbf{Y}} P_y(y_{\text{pred}})]$ as the interval for the probability that this prediction is correct. If this interval is denoted by $[a, b]$, the complementary interval $[1 - b, 1 - a]$ is called the error probability interval, and its ends $1 - b$ and $1 - a$ are referred as lower error probability and upper error probability, respectively. The details of Venn probability machine are presented in Algorithm 2.

A useful property of Venn probability machine is its self-calibration nature in the on-line learning setting. It has been proved that the probability bounds generated by Venn probability machines is well-calibrated [29]. They bound the true conditional probability for each new test object in an on-line setting. Using the upper and lower bounds generated by Venn probability machines for conditional probabilities, we can reliably estimate the bounds for the number of errors made.

3.3 Region prediction for Venn probability machine

At each step Venn probability machines can provide one prediction corresponding to the highest probability. On the other hand, Conformal Predictors can output region predictions. To compare the two algorithms we want to convert probabilistic predictions of Venn Probability Machine into region predictions. Therefore, a simple method is developed to convert probability forecasts of Venn probability machines into region predictions and it is then possible to compare the performance of conformal predictors and Venn probability machines.

Algorithm 2 Venn probability machine algorithm

Require: training examples $\{(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})\}$, label space $\mathbf{Y}$, new example x_n,
Require: taxonomy A_n.
for $y = 1$ to $|\mathbf{Y}|$ **do**
combine (x_n, y) with the training examples (We now have n examples $\{(x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), (x_n, y)\}$)
partition the n examples into categories T, assigning (x_i, y_i) and (x_j, y_j) to the same category if and only if $A_n(\wr z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, (x_n, y) \int, z_i) = A_n(\wr z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, (x_n, y) \int, z_j)$
for $i = 1$ to $|\mathbf{Y}|$ **do**
$F_{y,i} = \frac{|\{(x^*, y^*) \in T : y^* = y\}|}{|T|}$
end for
end for
$Q_i = \arg\min_{y \in \mathbf{Y}} F_{y,i}$
$i_{best} = \arg\max_{i \in \mathbf{Y}} Q_i$
predict $\hat{y} = \arg\max_{y \in \mathbf{Y}} F_{y,i_{best}}$
Output: predicted probability interval for $\hat{y}$ as $[\min_{y \in \mathbf{Y}} F_{y,\hat{y}}, \max_{y \in \mathbf{Y}} F_{y,\hat{y}}]$
Output: error probability interval $[1\text{-}\max_{y \in \mathbf{Y}} F_{y,\hat{y}}, 1\text{-}\min_{y \mathbf{Y}} F_{y,\hat{y}}]$
Output: predicted mean probability for $\hat{y}$ as $\text{mean}_{y \in \mathbf{Y}} F_{y,\hat{y}}$

Intuitively, we use the fact that the probability predictions are estimates of conditional probabilities and also assume that labels are mutually exclusive. Therefore, summing these predictions becomes a conditional probability of a conjunction of labels, which can be used to choose the labels to include in the region predictions at the desired confidence level. Consider the case when a Venn probability machine outputs a sequence of probability forecasts $p_1, p_2, \ldots, p_l$, where p_i is the probability that label i is correct. Let us consider the case where $p_1 > p_2 > \cdots > p_l$. If it is not the case, we can always rename the labels. Now to obtain region prediction we carry out the following procedure: we output all labels i such that $\sum_{j=1}^{i} p_j \leq 1 - \delta$, where δ is the significance level, i.e. the level of mistakes we can tolerate, as a possible label. For the label n such that $\sum_{j=1}^{n-1} p_j \leq 1 - \delta$, but $\sum_{j=1}^{n} p_j > 1 - \delta$ we need a more sophisticated approach: we will output label n as a possible label if and only if $(\text{rand}() > (1 - \delta - \sum_{j=1}^{n-1} p_j)/p_n)$, where rand() is a random number between 0 and 1. The reason for this condition is that we want the probability for a label from the output to be correct with a predefined probability $1 - \delta$. A more rigorous description of these steps is given in Algorithm 3.

We have presented how confidence region predictions can be obtained. We want that the obtained confidence region predictions are well calibrated, i.e. the fraction of errors at

confidence level $1-\delta$ should be at most δ. We define $\mathtt{Err}_N^{1-\delta}$ as the averaged errors over all the N test examples. It is possible to prove that this method of converting probability forecasts into confidence region predictions described above are well-calibrated for a classifier that outputs Bayes optimal forecasts as shown below.

Theorem 1 *Let* $\mathbf{P}$ *be a probability distribution on* $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$. *If a classifier's forecasts are Bayes optimal in the sense that* $\hat{P}(y_i = j|\mathbf{x}_i) = \mathbf{P}(y_i = j|\mathbf{x}_i)$ *for all* $j \in \mathbf{Y}$, $1 \leq i \leq N$, $\epsilon \in (0, 1)$, *then* $\mathbf{P}^N(\mathtt{Err}_N^{1-\delta} \geq \delta + \epsilon) \leq e^{-2\epsilon^2 N}$

Algorithm 3 Region predictions for Venn probability machines

Require: significance level $\delta > 0$
Require: probability forecasting $(p_1, p_2, \ldots, p_l) : p_1 > p_2 > \cdots > p_l$
$P = 0$
$i = 1$
EXIT_FLAG $= 0$
while $(P < 1-\delta)$ AND (EXIT_FLAG $==0$) **do**
  if $P + p_i > 1 - \delta$ **then**
    EXIT_FLAG $= 1$
    if $(rand() > (1 - \delta - P)/p_i)$ **then**
      add label i to the set of predictions
    end if
  else
    $P = P + p_i$
    add label i to the set of predictions
  end if
  $i = i + 1$
end while

This means if the probability forecasts are Bayes optimal, then even for small finite sequences the confidence region predictors will be well-calibrated with high probability. This proof follows directly from the Azuma-Hoeffding inequality [8] and thus justifies this method for generating confidence region predictions.

4 Experiments and results

Both conformal predictors and Venn probability machines can be evaluated in *on-line learning setting* where the learning algorithm is presented with each unlabelled example in sequence. It is informed of the true label of the example after it has made its prediction, but before the next unlabelled example is presented [29]. As an on-line learning experiment progresses, the knowledge available to the learning algorithm from prior experience increases. We would therefore expect performance to improve as the experiment progresses. There are two main measures of performance for region prediction, namely validity and efficiency [29]. Validity means that the probability of the output being correct is close to the confidence level. High efficiency is achieved when the output of a prediction algorithm consists of a number of outcomes close to one. Ideally, we are interested in valid efficient predictors, i.e. algorithms which output a small number of outcomes with a guarantee that one of these outcomes will happen in real life. In this paper both types of performance measures for reliable predictors are considered for conformal predictors and Venn probability machines in the on-line learning.

4.1 Datasets

The network traffic datasets are generated from both link directions on Genome campus with three institutions on site that employ about 1,000 researchers, administrators and technical staff. The datasets cover a full 24 hour period for a weekday. The collection is comprised of 11 datasets (entry01, entry02, entry03, entry04, entry05, entry06, entry07, entry08, entry09, entry10 and entry12) and is publicly available at http://www.cl.cam.ac.uk/research/srg/netos/nprobe/data/papers/sigmetrics/index.html. This collection of datasets has been analysed in a number of publications [7,19,23]. Each network flow sample is obtained from a complete bidirectional TCP flow and has 248 attributes including source and destination port numbers [25]. Flows are labelled according to the type of traffic they represent and eight types of network applications are considered, see Table 1 [25]. These datasets have been used in a number of papers [1,19] and similarly a subset of the features (10 features) are chosen and these features are used in the following experiments (for example, see Table 1 in [18] for the list of these features).

4.2 Underlying algorithms

Both conformal predictors (CP) and Venn probability machines (VPM) are generic frameworks. To build a particular conformal predictor or Venn probability machine, an underlying machine learning algorithm is needed to specify a nonconformity measure or a Venn taxonomy. Virtually any algorithm can be used to construct a Venn taxonomy or a nonconformity measure. However, as Venn Probability Machines and Conformal Predictors are used in the on-line setting in our experiments, the underlying algorithms are preferred to be computationally efficient. Therefore, we consider Nearest Neighbours and Nearest Centroid as the underlying algorithm in our experiments.

The Nearest Neighbours (NN) algorithm is one of the simplest methods in machine learning for making predictions. It uses the idea that close objects (according to a distance measure) are likely similar, and as such, probably belong to the same class. In the simplest case, this algorithm finds the current object's nearest neighbour using a predefined distance function $d(x_i, x_j)$ (the Euclidean distance measure is commonly used) and assigns its label to the current example.

Table 1 Traffic classes

Traffic class	Example applications	Label
WWW	http, https	1
MAIL	imap, pop2/3, smtp	2
FTP	ftp	3
ATTACK	portscan, worms, viruses	4
P2P	napster, eMule, gnutella, eDonkey	5
DATABASE	mysql, dbase, Oracle, SQLnet	6
MULTIMEDIA	realmedia, windows mediaplayer	7
SERVICES	X11. dns, ident, ldap, ntp	8

This algorithm can be expanded to the case of several nearest neighbours. The nonconformity measure for the example $z_i = (x_i, y_i)$ in the 1-nearest neighbour algorithm setting can be defined as follows:

$$\alpha_i = \frac{\min_{y_j = y_i} d(x_j, x_i)}{\min_{y_j \neq y_i} d(x_j, x_i)}.$$

In the case of k nearest neighbours, the nearest neighbours' mean value is taken instead of x_j. Again the k nearest neighbours with the same label as (x_i, y_i) are selected to calculate the upper term of the fraction, and those with different labels to calculate the lower term of the fraction.

Two types of taxonomies based on the NN algorithm can be used for Venn probability machines. One type, called "voting" taxonomies, says that two objects belong to the same category if and only if for each object the most frequent label among its k nearest neighbours is the same. The second type of taxonomies, called "all info" taxonomies, says that two objects belong to the same category if and only if for each of them the ordered sequence of its k nearest neighbours' labels is the same.

Unlike the NN algorithm, the Nearest Centroid (NC) algorithm (see [2]) does not require computing the distances between each pair of objects. Instead, it measures the distance between an object and the centroid (i.e. the average) of all objects belonging to the same class. For Venn probability machines, two objects belong to the same category if and only if their nearest centroids coincide. For conformal predictors the nonconformity measure can be calculated for each example (x_i, y_i) as follows:

$$\alpha_i = \alpha_i(x_i, y_i) = \frac{d(\mu_{y_i}, x_i)}{\min_{y \neq y_i} d(\mu_y, x_i)},$$

where μ_y is the centroid of all objects with label y.

4.3 Results

In the offline learning setting, we use the"entry01" as the training set and make classification on all other 10 datasets. Table 2 shows the VPM performance on 3 datasets ("entry07", "entry09" and "entry12") using "all info" taxonomy in comparison with the NN algorithm. The same datasets have been analysed using the Naive Bayes estimator and its performance is similar to that of kNN [23]. For example, it is reported that the best average percentage of accuracy of Naive Bayes performed on the selected features is 96.29 %.

However, the main focus of the paper is the performance of conformal predictors and Venn probability predictors evaluated in the on-line setting. Table 3 shows 3 example p-values produced by CP (based on 3NN) on the dataset "entry06". Table 4 illustrates the corresponding single predictions with their confidence and credibility measure. Confidence is taken as 1 minus the second largest p-value and credibility is just the largest p-value. These confidence and credibility values provide additional information about the predictions. High confidence (close to 100 %) shows that all alternatives to the predicted label are unlikely. Low credibility indicates that the whole situation is suspect: this example looks unusual as compared with the examples in the training set.

As discussed earlier, the performance of these predictors is measured by validity (measured in error rates) and efficiency (measured in average number of labels observed). Validity means that the error rate is close to the expected value, which is equal to a predefined constant (significance level). It is important to note that conformal predictors provide guaranteed validity. Efficiency reflects how useful the predictions output by a predictor are. Ideally we would like our prediction system to output only one label at each step of predictions. The validity property of Venn probability machines is expressed differently from the same property of conformal predictors due to the difference in the output. As discussed in Sect. 3.3, we can convert the probability forecasts of Venn probability machines into confidence region predictions and then compare the results. For illustration purpose, significance levels of 1, 5 and 10 % were used. Tables 5 and 6 provide a summary of the experimental results of conformal predictors (CP) and Venn probability machines (VPM) on the datasets "entry01" and "entry02" with different number of nearest neighbours (subject to statistical variation). In these tables, VPM-V stands for Venn Probability Machine based on "voting" taxonomy and VPM-AI stands for Venn Probability Machine based on "all info" taxonomy. Note that

Table 2 Performance of VPM (accuracy %—using all info taxonomy)

Alg.	entry07		entry09		entry12	
	NN	VPM	NN	VPM	NN	VPM
1NN	96.19	99.25	94.89	97.97	96.78	97.49
2NN	96.43	99.22	95.21	98.38	96.41	98.24
3NN	96.45	99.26	95.12	98.41	96.29	95.52
4NN	97.90	99.22	96.68	98.45	96.31	95.53
5NN	98.00	99.21	96.69	98.24	96.76	96.18
6NN	98.00	99.35	96.75	98.39	96.52	96.36
7NN	98.01	99.32	96.71	98.40	97.00	96.34
8NN	97.98	99.03	96.69	98.27	96.83	96.05
9NN	98.00	99.02	96.71	98.20	96.85	96.17
10NN	97.98	99.00	96.69	98.27	96.83	95.95

Table 3 *P* values calculated by CP

1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	6 (%)	7 (%)	8 (%)
0.1	0.1	60.4	0.1	0.1	0.1	0.1	0.2
0.7	12.2	0.1	0.8	0.1	0.1	0.3	0.8
0.6	0.6	0.1	0.4	0.1	0.1	0.1	20.3

Table 4 Single prediction by CP

True label	Predicted label	Confidence (%)	Credibility (%)
3	3	99.8	60.4
2	2	99.2	12.2
8	8	99.4	20.3

the algorithm tries to match the given error rate and even if with outputting only one label it can achieve a lower error rate it will sometimes refrain from outputting any labels at all so that the total error rate is closer to the parameter of the algorithm. This can be considered as the strength of the algorithm as in applications one can always select the label with the highest p-value as the output and thus reduce the error rate. Some of the main conclusions which can be drawn from the results are as follows:

- Venn probability machines and conformal predictors have similar performance, both from a validity and an efficiency point of view.
- Venn probability machines in the multiple prediction mode provide empirical validity, although not as well as conformal predictors.
- The efficiency of the algorithms changes depending on the dataset.
- The number of nearest neighbours does not significantly affect the performance of the Nearest Neighbours algorithm.
- On the datasets used in this paper, Nearest Centroid does not perform as well as the nearest neighbours algorithms.

Figure 1 shows an example of experimental results for Venn Probability Machines on "entry01" dataset in the on-line setting. Venn Probability Machines here are run in the single prediction mode. It can be seen that in the long run the total number of errors lies between the cumulative lower and upper bounds and this property of Venn Probability Machines guarantees the prediction quality. A decision making system which employs a Venn Probability Machine can use this property to improve its performance, for example by ignoring a prediction if such probability is too low.

Figure 2 represents a classical experimental result for Conformal Predictors. The graph of the number of errors over time should be a straight line and the percentage of errors at each point should approximately equal to the significance level. In many applications we can estimate a threshold on the number of errors we can tolerate. Conformal Predictors is a guaranteed method (i.e. there exist a theorem saying that the performance is guaranteed, see [29]) of making predictions with a predefined number of errors we can tolerate. As it can be seen from Fig. 2 the plots of the number of errors are not straight lines, this is due to statistical fluctuations.

Figure 3 shows one example of experimental results of efficiency for conformal predictors for different significance

Table 5 Performance comparison on dataset "entry01"

Pred. Alg.	Used Alg.	Error rate (%)			Average no. of labels		
		1 %	5 %	10 %	1 %	5 %	10 %
CP	1NN	1.13	4.88	6.25	1.38	1.34	1.33
	2NN	1.10	4.89	6.67	1.38	1.34	1.33
	3NN	1.12	4.79	6.83	1.38	1.34	1.32
	4NN	1.16	4.88	6.94	1.38	1.34	1.32
	5NN	1.19	4.85	7.02	1.38	1.34	1.32
	6NN	1.17	4.94	7.11	1.38	1.34	1.32
	7NN	1.17	4.84	7.23	1.38	1.34	1.32
	8NN	1.16	4.92	7.35	1.36	1.33	1.30
	9NN	1.16	4.90	7.47	1.36	1.32	1.30
	10NN	1.17	4.92	7.54	1.36	1.32	1.29
CP	NC	0.53	0.53	9.26	7.10	7.10	4.50
VPM-V	1NN	1.19	4.96	10.20	1.06	0.96	0.90
	2NN	0.93	4.91	10.05	1.06	0.96	0.90
	3NN	1.09	4.81	10.11	1.07	0.96	0.90
	4NN	0.95	5.30	9.83	1.07	0.96	0.91
	5NN	1.00	4.89	9.27	1.09	0.96	0.91
	6NN	0.82	5.12	9.16	1.09	0.97	0.91
	7NN	0.94	4.81	9.91	1.09	0.97	0.91
	8NN	1.19	5.09	9.77	1.09	0.97	0.91
	9NN	1.13	5.47	10.32	1.09	0.97	0.90
	10NN	1.01	5.62	9.75	1.09	0.97	0.91
VPM-AI	1NN	1.19	4.96	10.20	1.06	0.96	0.90
	2NN	0.99	4.94	10.09	1.05	0.96	0.91
	3NN	1.09	4.78	10.00	1.04	0.96	0.91
	4NN	0.95	5.32	9.81	1.04	0.96	0.91
	5NN	0.75	3.15	5.74	1.38	1.31	1.25
	6NN	0.75	3.27	6.06	1.38	1.31	1.25
	7NN	0.74	3.25	6.40	1.38	1.31	1.24
	8NN	0.83	3.37	6.08	1.37	1.27	1.17
	9NN	0.88	3.52	6.61	1.38	1.31	1.24
	10NN	0.75	3.67	6.27	1.38	1.27	1.17
VPM	NC	1.09	4.71	10.20	3.32	1.94	1.64

levels. It can be seen that the conformal predictors can ensure validity by giving a region prediction. Instead of predicting a single class label for a new example, it can suggest several, any of which may be correct. Higher confidence levels can be accommodated by giving less efficient region predictions to hedge the classifier's decision.

5 Discussions and conclusions

Reliable estimation of confidence remains a significant challenge as learning algorithms proliferate into many challenging real world applications. This paper was devoted to classification algorithms with on-line validity and presented two recently developed methods: conformal predictors and Venn probability machines. The performance of these two algorithms was assessed in terms of validity and efficiency on publicly available real network traffic datasets in the on-line setting. For conformal predictors, validity implies that the number of errors is close to the preset significance level and efficiency means predicting as few as possible multiple predictions. For Venn probability machines, validity results in output probability distributions agreeing with observed frequencies. Venn probability machine is efficient if the outputted probability interval is narrow and close enough to 1. The application of these two methods to network traffic classification problems demonstrated that the methods are efficient while maintaining the property of validity.

Table 6 Performance comparison on "entry02"

Pred. Alg.	Used Alg.	Error rate (%) 1 %	5 %	10 %	Average no. of labels 1 %	5 %	10 %
CP	1NN	0.96	5.18	7.58	1.90	1.85	1.83
	2NN	0.98	5.16	7.87	1.89	1.85	1.82
	3NN	0.96	5.22	8.06	1.88	1.84	1.81
	4NN	1.01	5.22	8.14	1.88	1.84	1.81
	5NN	1.05	5.22	8.19	1.88	1.84	1.81
	6NN	1.03	5.24	8.24	1.88	1.84	1.81
	7NN	1.02	5.23	8.31	1.88	1.84	1.81
	8NN	1.02	5.27	8.38	1.88	1.84	1.81
	9NN	1.02	5.29	8.46	1.84	1.80	1.77
	10NN	1.00	5.23	8.53	1.66	1.62	1.59
CP	NC	0.50	4.83	7.80	6.68	6.31	2.32
VPM-V	1NN	0.83	5.00	9.95	1.00	0.95	0.90
	2NN	1.20	5.11	10.07	0.99	0.95	0.90
	3NN	1.21	4.85	9.88	1.00	0.96	0.90
	4NN	1.08	4.99	9.72	1.00	0.95	0.91
	5NN	1.06	5.18	10.10	1.01	0.95	0.90
	6NN	1.10	5.03	9.58	1.00	0.95	0.91
	7NN	1.03	5.45	9.61	1.01	0.95	0.91
	8NN	0.98	5.00	10.22	1.01	0.96	0.90
	9NN	0.92	5.07	10.40	1.01	0.96	0.90
	10NN	0.91	5.08	9.82	1.01	0.96	0.91
VPM-AI	1NN	0.83	5.00	9.95	1.00	0.95	0.90
	2NN	1.22	5.11	10.12	0.99	0.95	0.90
	3NN	1.26	4.88	9.91	0.99	0.96	0.91
	4NN	1.07	5.05	9.76	1.00	0.95	0.91
	5NN	1.07	5.26	10.12	1.00	0.95	0.90
	6NN	1.10	5.02	9.53	1.00	0.96	0.91
	7NN	1.05	5.43	9.61	1.00	0.95	0.91
	8NN	0.96	5.03	10.21	1.00	0.95	0.90
	9NN	0.98	5.08	10.41	1.00	0.96	0.90
	10NN	0.90	5.11	9.90	1.00	0.95	0.91
VPM	NC	1.01	4.97	10.32	2.85	1.90	1.50

We can compare both conformal predictors and Venn probability machines with other existing approaches. In particular, we consider two main classes of algorithms: simple predictors and probability predictors. A simple predictor such as support vector machine [28] and decision tree only produces a single label and probability forecasting such as logistic regression and naive Bayes gives one probability distribution. In contrast to these simple predictors and probability forecasting, conformal predictors and Venn probability machines hedge predictions, i.e., express how much a user can rely on them.

When considering simple predictions, we can also use notions of validity and efficiency. In this respect, simple predictors and conformal predictors demonstrate opposite approaches to learning. This is summarised in Table 7. In simple predictions, the efficiency is guaranteed since each prediction is singleton, but validity is not. In conformal predictors, the validity of predictions is guaranteed, but efficiency depends on the nonconformity measure which is often defined using an underlying algorithm. This has been illustrated in our experimental results (see Tables 5, 6). In addition, conformal predictors allow us to balance efficiency and the error rate: lower preset error rates produce larger region predictions, and vice versa. This feature makes conformal predictors a flexible tool. The Internet continually evolves in scope and complexity, much faster than our ability

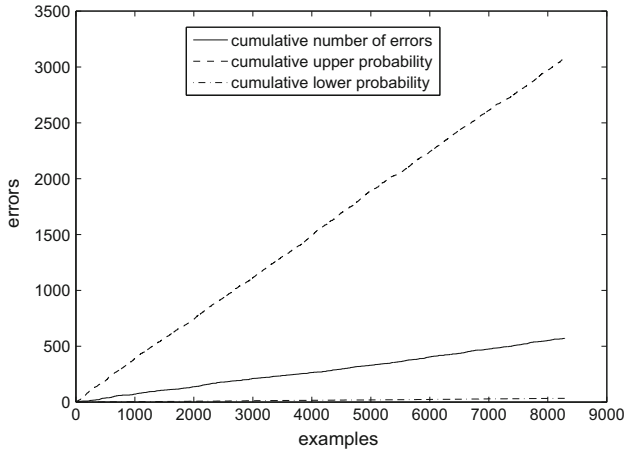

Fig. 1 Cumulative number of errors and upper and lower bounds (entry01, 5 Nearest Neighbours, "all info" taxonomy, single prediction)

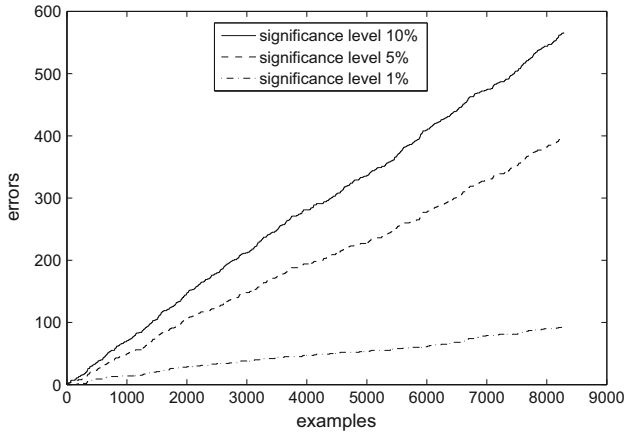

Fig. 2 Number of errors for different significance levels (entry01, 3 Nearest Neighbours, multiple predictions)

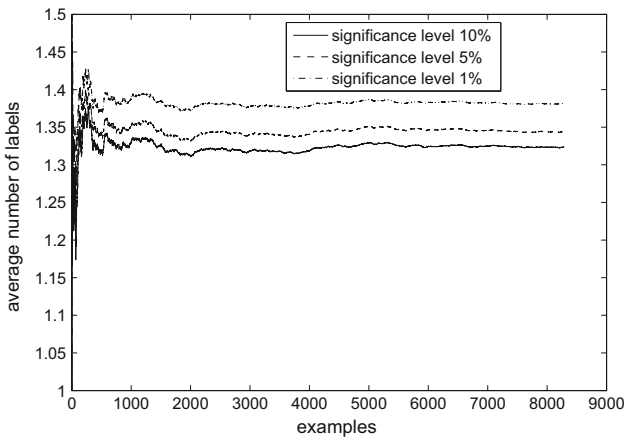

Fig. 3 Average number of labels for different significance levels (entry01, 3 Nearest Neighbours, multiple predictions)

Table 7 Comparison of conformal predictor and Venn probability machine with simple predictor and probability forecasting

Predictor	Simple predictor	Conformal predictor
Prediction output	Single prediction	Region prediction
Validity	Depends on the algorithm	Guaranteed
Efficiency	Guaranteed	Depends on the nonconformity measure
Predictor	Probability forecasting	Venn probability machine
Prediction output	Single distribution	Multiprobability distributions
Validity	Guaranteed under strong statistical assumption	Guaranteed
Efficiency	Depends on the algorithm	Depends on the Venn taxonomy

to characterise, understand, control or predict it. Dynamic resource allocation with the support of traffic prediction can efficiently utilise the network resources and support Quality of Service. One of the key requirements for dynamic resource allocation framework is to predict traffic in the next control time interval based on historical data and online measurements of traffic characteristics over appropriate timescales. The novelty of Conformal Predictor and Venn Probability Machine is that they can learn and predict simultaneously, continually improving their performance as they make each new prediction and ascertain how accurate it is. Both Conformal Predictor and Venn Probability Machine could form a significant component of any resource management mechanism, where decisions need to be taken in a stochastic environment. These accuracy reliability measures could allow service provider and network carrier to choose appropriate allocation strategies by identifying anticipated resource demands and placing connections accordingly. For example, these actions could be tempered by the strength of the prediction, allowing the resource management process to effectively avoid potential "hot spot" and thus successfully accommodate more traffic.

Algorithms with on-line validity represent a flexible framework and can use practically any known machine learning method as underlying algorithm for designing a nonconformity measure or a Venn taxonomy. The performance of conformal predictors and Venn probability machines is usually in line with the accuracy of the underlying algorithm. Therefore, it would be beneficial to deploy new underlying algorithms in order to design new nonconformity measures and Venn taxonomies to inherit their ability to perform well on certain types of data.

Acknowledgments This work was supported by EPSRC through grant EP/E000053/1, "Machine Learning for Resource Management in Next-Generation Optical Networks".

References

1. Auld, T., Moore, A.W., Gull, S.: Bayesian neural networks for internet traffic classification. IEEE Trans. Neural Netw. **18**(1), 223–239 (2007)
2. Bellotti, T., Luo, Z., Gammerman, A., van Delft, F.W., Saha, V.: Qualified predictions for microarray and proteomics pattern diagnostics with confidence machines. Int. J. Neural Syst. **15**(4), 1–12 (2005)
3. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines (and other Kernel based learning methods). Cambridge University Press, Cambridge (2000)
4. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis. Chapman and Hall/CRC, London (2003)
5. Dashevskiy, M., Luo, Z.: Reliable probabilistic classification of internet traffic. Int. J. Inf. Acquis. **6**(2), 133–146 (2009)
6. Dashevskiy, M., Luo, Z.: Predictions with confidence in applications. In: The 6th international conference on machine learning and data mining (MLDM 2009), pp. 775–786 (2009)
7. Dashevskiy, M., Luo, Z.: Reliable probabilistic classification and its application to internet traffic. ICIC (1), 380–388 (2008)
8. Devroye, L., Gyorfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, New York (1996)
9. Erman, J., Mahanti, A., Arlitt, M.: Traffic classification using clustering algorithms. In: Proceedings of the 2006 SIGCOMM workshop on Mining network data, pp. 281–286 (2006)
10. Este, A., Gringoli, F., Salgarelli, L.: Support vector machines for TCP traffic classification. Comput. Netw. **53**(14), 2476–2490 (2009)
11. Floyd, S., Warmuth, M.K.: Complete compression, learnability, and the Vapnik-Chervonenkis dimension. Mach. Learn. **21**, 269–304 (1995)
12. Gammerman, A., Vovk, V.: Hedging predictions in machine learning: the second computer journal lecture. Comput. J. **50**(2), 151–163 (2007)
13. Gu, C., Zhang, S., Xue, X.: Encrypted internet traffic classification method based on host behavior. Int. J. Digit. Content Technol. Appl. **5**(3), 167–174 (2011)
14. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: BLINC: Multilevel traffic classification in the dark. In: Proc. of ACM SIGCOMM, pp. 229–240 (2005)
15. Kim, H., Claffy, K.C., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Proceedings of the 2008 ACM CoNEXT Conference, New York, USA, Article 11 (2008)
16. Korb, K.B.: Calibration and the evaluation of predictive learners. In: Proceedings of sixteenth international joint conference on artificial intelligence, pp. 73–77 (1999)
17. Lambrou, A., Papadopoulos, H., Gammerman, A.: Reliable confidence measures for medical diagnosis with evolutionary algorithms. IEEE Trans. Inf. Technol. Biomed. **15**(1), 93–99 (2011)
18. Li, W., Abdin, K., Dann, R., Moore, A.: Approaching real-time network traffic classification. Technical Report, RR-06-12, Dept of Computer Science, Queen Mary, University of London (2006)
19. Li, W., Canini, M., Moore, A.W., Bolla, R.: Efficient application identification and the temporal and spatial stability of classification schema. Comput. Netw. **53**(6), 790–809 (2009)
20. Littlestone, N., Warmuth, M.K.: Relating data compression and learnability. Technical report, University of California, Santa Cruz (1986)
21. Madhukar, A., Williamson, C.: A Longitudinal Study of P2P Traffic Classification. In: Proceedings of the 14th IEEE international symposium on modeling, analysis, and simulation (MASCOTS'06), pp. 179–188 (2006)
22. Mitchell, T.M.: Machine Learning. McGrow-Hill, New York (1997)
23. Moore, A.W., Zuev, D.: Internet traffic classification using Bayesian analysis techniques. SIGMETRICS, 50–60 (2005)
24. Moore, A.W., Papagiannaki, D.: Toward the accurate identification of network applications. In: Proceedings of the sixth passive and active measurement workshop, pp. 50–60 (2005)
25. Moore, A.W., Zuev, D., Crogan, M.: Discriminators for use in flow-based classification, Technical Report RR-05-13, Dept. of Computer Science, Queen Mary, University of London (2005)
26. Papatheocharous, E., Papadopoulos, H., Andreou, A.S.: Software Effort Estimation with Ridge Regression and Evolutionary Attribute Selection. In: Proceedings of the 3rd workshop on artificial intelligence techniques in software engineering (2010)
27. Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of P2P traffic using application signatures. In: Proceedings of the 13th international conference on world wide web (WWW' 04), New York, USA, pp. 512–521 (2004)
28. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
29. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, Berlin (2005)
30. Yoon, S.-H., Park, J.-W., Park, J.-S., Oh, Y.-S., Kim, M.-S.: Internet application traffic classification using fixed IP-port. LNCS, vol. 5787, pp. 21–30 (2009)
31. Zander, S., Nguyen, T., Armitage, G.: Automated traffic classification and application identification using machine learning. In: The IEEE Conference on Local Computer Networks, pp. 250–257 (2005)