Multivariate Data Fusion-Based Learning of Video Content and Service Distribution for Cyber Physical Social Systems

Sudip Misra, *Senior Member, IEEE*, Sumit Goswami, and Chaynika Taneja

***Abstract*— Integration of physical processes with the computing world is driving newer challenges for networking frameworks. Cyber physical social systems (CPSSs) are another upcoming paradigm that encompasses the ever-growing interaction between the physical, social, and cyber worlds. As communication networks form the basis of these interactions, a cognitive evaluation of networks is called for. This CPSS driven network evolution was a direction motivating this paper. With the implementation of the next generation networks, traffic from real-time interactive services, such as video conferencing, is surpassing those of conventional transactional services. As such multimedia data transportation over IP networks has stringent quality constraints in terms of required bandwidth, latency, and jitter, legacy networks with no quality of service face challenges in terms of performance. We attempt to perform a multivariate analysis of video call record data collected from a wide area organizational network over a period of time. Learning-based prediction is attempted by training four classifiers: naïve Bayes, k-nearest neighbor, decision tree, and support vector machine. Two independent set of experiments were conducted with oversights of bandwidth and destination prediction. Both the discrete and continuous valued predictors were involved in the training. Performance evaluation of the generated hypothesis in both the cases was conducted using tenfold cross validation. Combined analysis using the assorted combinations of attributes was conducted, and thereafter, the effect of each feature was evaluated through singular attribute portioning. This paper presents observations, which exhibit deviations from the conventional machine learning paradigms. An attempt to increase the prediction accuracy of the classifiers was made through the boosting ensemble methodology. However, miniscule addition in performance was achieved. A maximum prediction accuracy of 81% for bandwidth and 60% for destination was obtained. Reasons of low accuracy of conventionally better performing algorithm were reasoned with a mathematical comprehension. Divergence of the obtained results from the accepted patterns poses an open research problem, particularly with respect to the nature and peculiarities of the data set. The proposed learning technique can have potential applications in social, tactical, and strategic spheres.**

***Index Terms*— Classifier, cyber physical social system (CPSS), data fusion, decision tree, k-nearest neighbor (k-NN), machine learning, naïve Bayes, service distribution, stylometrics, support vector machine (SVM), video conferencing.**

Manuscript received June 16, 2015; revised February 17, 2016; accepted April 14, 2016. Date of publication May 23, 2016; date of current version June 17, 2016. Some parts of the work reported in this paper were presented at the IEEE Advanced Networks and Telecommunication Systems 2014, New Delhi, India, and the IEEE International Conference on Communications 2015, London, U.K.

S. Misra and S. Goswami are with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur 721302, India (e-mail: sudipm@iitkgp.ac.in; sgoswami@iitkgp.ac.in).

C. Taneja is with the Defence Research and Development Organization, New Delhi 110054, India (e-mail: chaynika@hqr.drdo.in).

Digital Object Identifier 10.1109/TCSS.2016.2561200

I. Introduction

THE advent of new technologies in computing and communication has impacted human life in an unprecedented way. The paradigm of cyber physical social system (CPSS) signifies the impactful association of the physical, social, and computing worlds. Emergence of smart cars, cities, personal devices with continuous sensing, monitoring, and communication has necessitated the need for advanced and adaptive communication infrastructure. With applications becoming network centric, a reciprocal approach wherein the network has built-in learning to model and adapt itself to suit the application needs is required.

Video conference (VC), which uses the synchronized audio and video signals for instant two-way communication between two distant end points, is the most prevalent application over IP networks today. Increasing dependence on VC systems in a network has given rise to inquisitiveness about the quality of voice and video prior to connection establishment. For networks, which do not have predefined quality of service (QoS) parameters, the network criteria at the time of call setup are accompanied by connection adaptations. Source, destination, and bandwidth are the most common attributes for a video call in a network.

The establishment of connectivity between two points depends on a number of network parameters, which is primarily decided based on the selected source–destination pair as well as the available bandwidth. Bandwidth also decides the quality of a video call in terms of voice and picture quality, voice and video synchronization, jitters, freeze, and exchange of contents. Based on the conventional machine learning techniques, it can be safely assumed that the source and the destination can be predicted with fairly high degree of accuracy, given the intermediate network parameters. This hypothesis forms the basis of this paper of training classifiers for prediction of source–destination pairs and bandwidth with a fair degree of accuracy. A learning-based solution for this purpose is proposed.

Sections II–IV explain the motivation behind this paper and a detailed description of the data set. Experimental results along with mathematical analysis and discussions are presented in Sections V–VII. In Section VIII, we project a number of open research areas of immediate concern. This paper also poses critical research problems, as the results indicate significant deviation from the conventional theories of data mining and machine learning.

II. Related Work

Significant strides in computing and networking technologies are the driving force behind the evolution of CPSS systems. The inclusion of the social aspect to the physical and computing domains introduces new design and optimization challenges from network perspective. Significant work has been done using machine learning for an array of network applications ranging from anomaly detection to predictive modeling of resources. We have grouped the related work according to the challenge areas arising as a result of the CPSS driven network evolution as well as on the basis of the classifiers used for the learning.

A. Fault Tolerance, Link Quality, and Reliability

Link quality prediction in multimedia streaming was attempted by Jin *et al.* [1]. The significance of detecting the destination was felt in the work reported by Sarigiannidis *et al.* [2], wherein they devised a rule-based system to detect Sybil attack, where the detection paradigm was to locate two or more nodes in the same area of a wireless sensor network. Lindhorst *et al.* [3] attempted to accurately predict the failure of links in wireless mesh networks using the data mining techniques. They attempted to reduce the prediction latency by monitoring the parameters on the wireless local area network's media access control layer. Artificial neural networks was used for small-scale fading in wireless networks [4]. Their work could accurately predict Rician K factor in different terrains.

B. Resource and Service Management for CPSS

Resource and service management has a critical role in CPSS implementation to ensure guaranteed QoS and experience. Machine learning was used in variety of network related applications [5]–[8]. For an effective network support, classifiers was used for network application identification [9], end-to-end latency prediction [10], and network route determination [11], [12] in wired, ad hoc [13], as well as underwater networks [14].

Introduction of the social element in the cyber physical systems requires that the security and trust management on the underlying network is substantially robust. Auld *et al.* [15] used Bayesian neural networks for the identification of Internet traffic. Classification was done only using packet headers without any access to the payload. Anomaly detection using the Markov models was used to detect cyber attacks [16]. Historical data about computer events were used to model a normal profile and any deviation indicated an attack.

C. Training Classifiers

Naïve Bayes classifier and its improvisations were used for a variety of network related functions. Naïve Bayes along with AdaBoost was used for network anomaly intrusion detection by Li and Li [17] with very low false positive rate. Zhang *et al.* [18] used the classifier aggregated with a statistical bag-of-flow model for network traffic classification. The receiver operating characteristic curve generated by the naïve Bayes classifier was used to identify a drift in concept in use of network traffic [19]. This helped to reduce the false alarms in intrusion detection and prevention systems. Gumus *et al.* [20] proposed an online version of the classifier to discriminate against genuine and malicious network connections. The classifier was used to categorize the packets according to the QoS classes for suitable forwarding to respective virtual networks [21]. Kalva *et al.* [22] attempted image classification on the Web by combining the contextual information along with the naïve Bayes classifier. Support vector machine (SVM) along with Fisher discriminant analysis was used for detecting malicious sinks in wireless ad hoc networks [23].

D. Ensembling to Improve the Quality

Various researchers have proposed ensemble-based techniques to improve the classification results. Jun *et al.* [24] proposed the pairwise classification using gentle AdaBoost implemented using decision stumps. Pan *et al.* [25] proposed a novel graph-based ensemble boosting that could tackle imbalanced classes with inherent noise. Semisupervised learning boosted with AdaBoost was suggested to work on directly with unlabeled data by Xu *et al.* [26]. Chi and Porikli [27] proposed the use of multiple collaborative representations to achieve a gain in performance. Cross validation was used for optimal tuning of the regularization parameters. However, a comprehensive literature survey indicated limited research work had been carried out in the area of learning-based prediction with a direct correlation to the impact on the implementation of CPSS. Lack of sufficient data or data set with insignificant attributes can be the reasons for less coverage of this area.

A thorough analysis of the contemporary as well as the historical literature has revealed that though some results have been reported on multimedia streaming, which is a one-way communication and can withstand minor network delays, no work has reported on VC traffic, which is not only bandwidth intensive but also in real time, gets effected by reduction in throughput and can hog high network bandwidth continuously for a few hours. It was also observed in the literature that no experimentation was conducted in such a huge countrywide network as is done in this paper. Most of the reported experiments were conducted on data from a simulated environment, released in data competitions or lab/institute network. The behavior of an actual network with manifold increase in the size over experimental networks will definitely be different as has been clearly brought forward by the results in this paper. It also showed that the classifiers that performed best on experimental data did not give adequate performance with our data, but the other reportedly low-performing classifiers gave better results. To the best of our knowledge, no literature exists on source–destination prediction for any type of network, wired or wireless. Furthermore, a few studies attempted on bandwidth prediction are also based primarily on the source and destination, as the features and the effect of other attributes in isolation or in a certain combination were never reported for bandwidth analysis, which is another area in this paper.

III. Data Set

A. Network Test Bed

The video call records used for multivariate analysis in this paper were carried over a wide area network of a research

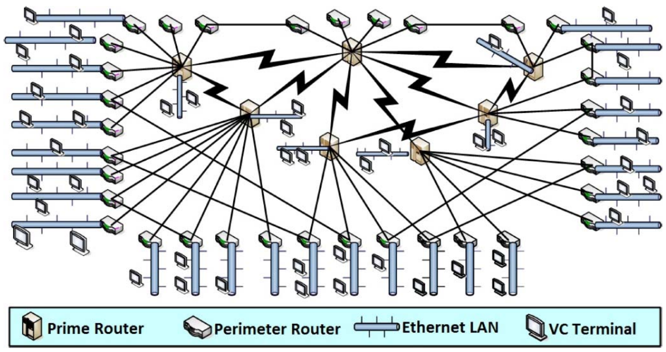


Fig. 1. Topological diagram of organizational intranet.

TABLE I
RANGE OF VALUES OF THE DATA SET ATTRIBUTES

Attribute	Range of values
Month	January to December
Year	2010, 2011, 2012, 2013
Hour	0-23
Source	110 entities
Destination	110 entities
Call type	Scheduled, ad hoc
Bandwidth	56-6144 kbps
Duration	1-798 minutes
Q.850 code	16 (call clearing), 0 (call drop)

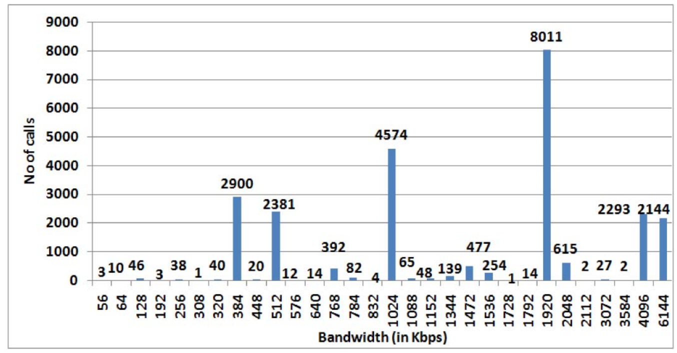


Fig. 2. Distribution of calls based on various bandwidths.

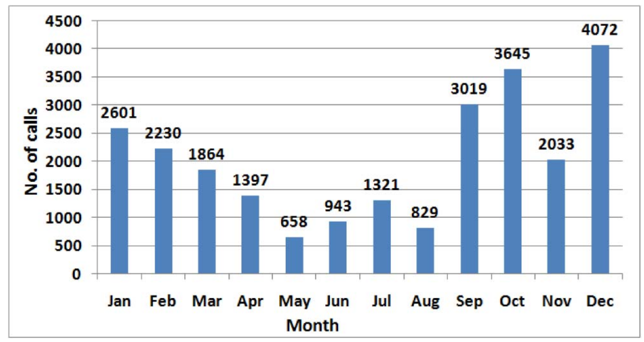


Fig. 3. Monthwise distribution of calls in the corpus.

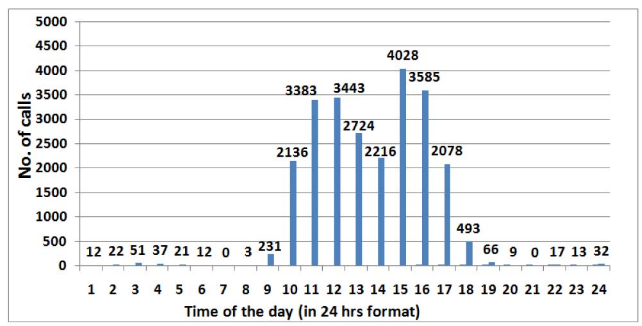


Fig. 4. Timewise distribution of calls in the corpus.

and development organization of the country. Fig. 1 shows the indicative topological representation of the point-to-point network test bed over which the observations were recorded. Partial redundancy between the prime routers is built in through links from alternative Internet service provider. The network does not have any QoS implemented in it. The VC traffic shares the network resources with other applications, such as email, Web site access, file upload, voice over IP, Web cast, collaboration, access to grid computing, access to private cloud, and database transactions. However, video traffic overshadows the percentage utilization of the bandwidth compared with other application data. Non-video traffic volume on the network is miniscule, as none of other applications generate bursty or heavy traffic. Hence, it can be theoretically assumed that adequate bandwidth is always available in the network for an unscheduled or impromptu VC connection.

B. Corpus Description

The experimental corpus used for carrying out the research work comprises of 24 865 VC call instances collected during September 2010 to December 2013. The raw data set was subjected to a pruning process leaving a corpus of 24 612 call records, which was used in this paper.

The attributes characterizing the instances were date of connectivity, time of connectivity, source, destination, call type, bandwidth, duration, and Q.850 cause code. The range of values of the attributes is given in Table I. The variation in the video call corpus in terms of bandwidth, month of call, and time of call is shown in Figs. 2, 3, and 4, respectively.

From the date of connectivity, only the month and the year of the connectivity are taken as the predictor. Similarly, from the time of the connectivity, only the hour of the connectivity is chosen to keep the model complexity under check. The rationale behind considering the month and the time of connectivity for machine learning is derived from the lack of uniformity exhibited by these features in the scatter plots against destination. Figs. 5 and 6 exhibit the scatter plots of month and hour of connectivity, respectively, against bandwidth. The limited uniformity in Figs. 5 and 6 acts as a significant attribute to add to the accuracy of the learning system.

IV. FEATURES

A detailed inspection of the data set brought out a few characteristic features, which are critical to the analysis and learning technique. While theoretically, only source and destination addresses govern the interconnecting bandwidth, the analysis of other parameters will add a convincing marker and contribute significantly in bandwidth prediction. The predictors under consideration are both of continuous and discrete nature. Call duration is an attribute that is

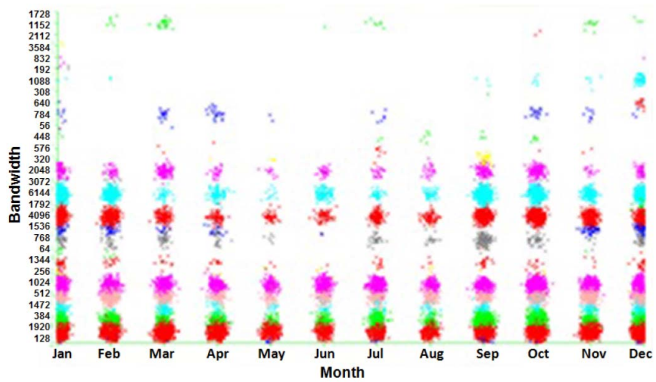


Fig. 5. Scatter plot of month versus bandwidth of call.

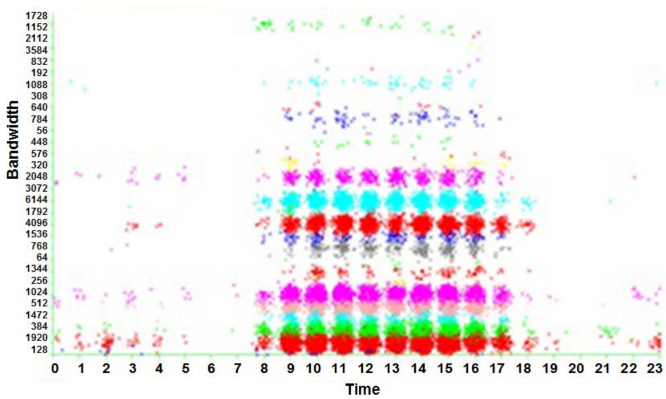


Fig. 6. Scatter plot of time versus bandwidth of call.

continuous valued. The ones with discrete nature are bandwidth, month of call, and time of call. Some attributes, such as source and destination addresses, Q.850 cause code, or call type, assume the values from closed defined data sets. These attributes contribute significantly in the stylometry, because the physical layer of the network is closely related to environmental conditions.

Most of the subnetworks still have copper wires in the last mile of connectivity. Intermediate devices, such as call manager, gateways, and network device, and their configurations, such as buffer space, can have representative impact on bandwidth selection. The choice of source and destination for learning-based prediction is conspicuous due to the direct relation. Hence, this forms the launching point for the set of experiments in this paper, thereafter augmenting it with other parameters. Learning with additional features is performed to analyze statistical self-sufficiency of each of these parameters. The scatter diagram of bandwidth versus month, as shown in Fig. 5, demonstrates very few high bandwidth calls in the month of February, June, and August due to rainfall.

Though the network under consideration is built of point-to-point leased circuits, subsequent transportation at the ISP level is over a single physical carrier wire employing multiplexing. The time of call is selected as a feature for learning, as the network conditions exhibit variability over a different time during a day. There are deep fades in the graph of the available bandwidth over a single channel at specific times of the day.

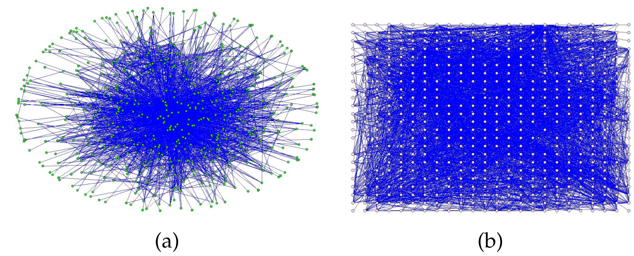


Fig. 7. Connectivity graph of source–destination video calls. (a) Nodes with low degree in the periphery. (b) Nodes with high degree.

This can be attributed to other shared users on the multiplexed channel conducting bandwidth intensive tasks, such as backup and replication. The scatter plot of bandwidth versus time shown in Fig. 6 affirms the observation of fewer number of high bandwidth calls during 11 A.M. and 2 P.M.

The call type attribute signifies whether the call is scheduled or *ad hoc*. Scheduled calls, which are carried out with prior notification to the call manager, are related to greater probability of getting connected at a higher bandwidth. This can be attributed to the manager's awareness about the upcoming call and subsequent blocking of resources for the call. *Ad hoc* call, wherein any user can call and connect to any other user in the existing network condition, has lower connection probabilities. Duration of the call is a natural feature choice, as it is assumed that high bandwidth connection for long durations is difficult to sustain. Long duration connectivity also leads to buffer overflow and call freeze.

A. Motivation

A visualization is modeled using all the source–destination pairs of the VC connectivity data set [28]. The source and the destination are chosen as the vertices and each connection is indicated by an edge. The network under consideration has 430 VC terminals, each capable of initiating a point-to-point VC connectivity with any other VC terminal. However, a panoramic view of the generated graph [5], as shown in Fig. 7(a), indicates a peculiar connectivity pattern. A subset of nodes are connected to only a few other VC systems. Another subset, as shown in Fig. 7(b), initiates connections with a larger number of the other VC systems. This style of connectivity, wherein a large number of nodes connect to only a small set of other nodes, is a potential momentous feature [29] in the detection of the group dynamics. This connectivity style when reinforced with other attributes, such as time and duration of connectivity, bandwidth, and call direction in a combined manner, can act as an important predictor. These attributes can be used for supervised learning of a model algorithm to predict the destination or bandwidth when other attributes are known.

The idea of determination of the effect of each attribute on the prediction accuracy of the model is challenging. The model can have significant utilization in tactical and social networks in anomaly detection, visualizing command and control, detecting change in the communication dynamics, detecting candidate cluster heads, and several other applications of commercial or strategic interest.

B. Classifiers Trained

We attempt to train classifiers using four different learning algorithms: naïve Bayes, k-nearest neighbors (k-NNs), decision tree, and SVM. A brief description of these classifiers is shown in the following.

C. Naïve Bayes

The Bayes theorem based on (1) relates the conditional and marginal probabilities of two events [30]

$$P(X|Y) = P(Y|X)P(X)/P(Y) \tag{1}$$

where

$P(X)$	prior or marginal probability of event X;
$P(Y)$	prior or marginal probability of event Y;
$P(X|Y)$	conditional probability of X given event Y occurs;
$P(Y|X)$	conditional probability of Y given event X occurs.

If the attribute vector is represented as F_i, where $i = 1, 2 \ldots, M$: M is the number of features or attributes used for classification.

The classes are represented by C_j, where $j = 1, 2 \ldots, K$: K is the number of classes into which the samples are to be classified. The posterior probability of a class given prior feature probability is given as [31]

$$P_r(C_j|F) = \frac{P(F|C_j)P(C_j)}{\sum_{i=1}^{k} P(F|C_j)P(C_j)}. \tag{2}$$

Naïve Bayes assumes that the attributes are independent. Hence, (2) reduces from multivariate analysis to univariate analysis [32] as

$$P_r(C_j|F) = K_{\text{prop}} P(C_j) \prod_{i=1}^{M} P(F_i|C_j) \tag{3}$$

where K_{prop} represents a constant of proportionality. The pseudocode of the algorithm is as follows [32].

1) Discretize training sample set.
2) Calculate prior probabilities $P_r(C_j) j = 1$ to K.
3) Estimate the conditional probabilities $P_r(F_i|C_j) i = 1$ to M, $j = 1$ to K.
4) For each sample point x_a from the training set, calculate posterior probability $P(C_j|F)$.
5) Assign x_a to class C_{assigned} if $C_{\text{assigned}} = \text{argmax}_j (C_j|F)$.

The effect of each attribute is calculated using Kononenko information gain formula [31] given by

$$\log_2 P_r = (\text{class}|\text{instance}) - \log_2 P_r(\text{class}). \tag{4}$$

The values indicate the effect of each predictor on the class probability.

D. k-Nearest Neighbor

k-NN is an instance-based supervised learning algorithm that classifies a new instance on the basis of majority to k-NNs. Different distance metrics can be used to calculate the k-NNs. Euclidean, Manhattan, or Minkowski distances [31] are used for numerical instances. Categorical attributes use Hamming distance

$$D_{\text{Euclidean}} = \sqrt{\sum_{i=1}^{k} (a_i - bi)^2} \tag{5}$$

$$D_{\text{Manhattan}} = \sum_{i=1}^{k} | \ a_i - bi \ | \tag{6}$$

$$D_{\text{Minkowski}} = \left[\sum_{i=1}^{k} (| \ a_i \ | - | \ bi \ |)^n \right]^{1/n} \tag{7}$$

$$D_{\text{Hamming}} = \begin{cases} 1, & \text{when } a_i \neq b_i \\ 0, & \text{when } a_i = b_i. \end{cases} \tag{8}$$

Cosine distance is also used in some implementations

$$D_{\text{Cosine}} = 1 - \langle a.b \rangle \langle a.b \rangle / \sqrt{\langle a.a \rangle \langle b.b \rangle} \tag{9}$$

where $\langle a.b \rangle$ is the dot product of the two vectors [32].

E. Decision Tree

A decision tree classifier graphically classifies a test input for given output class predictors. A tree is formed with splitting being done on the basis of attributes. The splitting is carried out recursively until a stopping criterion is reached. The fully developed tree is subject to pruning to remove splits that do not add significantly to performance. Pruning calculates the amount of error before and after each prune and attempts to minimize the error estimate Er_{prune} given by

$$Er_{\text{prune}} = x_{\text{miss}} + 1/(N + m_i) \tag{10}$$

where

x_{miss}	number of samples misclassified;
N	total number of samples in the training data set;
m_i	number of samples at the pruning stage i.

As compared with naïve Bayes and k-NN classifiers, decision trees are slightly unstable. This also means that the variance of the classifier is high as compared with the naïve Bayes and k-NN.

1) Splitting Decision Methodology: At each split, an information measure I is used for taking the splitting decision. At node n, the information measure is represented as $I(n)$. A decrease in this measure by splitting the node into its leaf children n_{left} and n_{right} is calculated. Mathematically, the reduction in information at a split S is represented by [32]

$$\Delta I(S, n) = I(n) - r_{\text{left}} I(n_{\text{left}}) - r_{\text{right}} I(n_{\text{right}}) \tag{11}$$

where r_{left} and r_{right} are the ratio, in which the nodes are divided into n_{left} and n_{right} at the split S.

Two parameters are used to measure the information at each node.

1) *Entropy:*

$$I(n) = \sum_{i=1}^{k} P(C_i|n)/\log P(C_i|n). \tag{12}$$

2) *Ginis Diversity Index:*

$$I(n) = 1 - \sum_{i=1}^{k} P^2(C_i|n). \tag{13}$$

F. Support Vector Machine

SVMs are a class of supervised learning methods that use a construction of a hyperplane to separate the classes. For a two-class case, the hyperplane, which is the decision boundary, is usually represented by

$$w^t X + b = 0 \tag{14}$$

where w is a weight vector and b is the intercept.

The support planes are given by

$$w^t X + b = 1 \tag{15}$$
$$w^t X + b = -1. \tag{16}$$

The margin between the two linearly separable classes is, therefore, given by $2/||w||$.

SVM attempts to maximize the margin. The optimization problem can, therefore, be considered as the minimization of $(||w||)^2/2$, subject to the condition that

$$w^t X + b \geq 1 \quad \text{for } X \epsilon X1 \tag{17}$$
$$w^t X + b \leq -1 \quad \text{for } X \epsilon X2. \tag{18}$$

Lagrangian multipliers are used to solve this optimization condition [33]

$$L(w, b, \alpha) = (||\ w\ ||)^2/2 - \sum_{i=1}^{n} \alpha_i [y_i(w^t X_i + b) - 1]. \tag{19}$$

The Karush–Kuhn–Tucker conditions provide the necessary conditions for this optimization problem. Differentiating (19) with respect to w and equating to zero, $\delta_L(w, b, \alpha)/\delta_w = 0$ gives

$$\sum_{i=1}^{N} y_i \alpha_i X_i = w. \tag{20}$$

Differentiating (19) with respect to b and equating to zero, $\delta_L(w, b, \alpha)/\delta_b = 0$ gives

$$\sum_{i=1}^{N} y_i \alpha_i = 0. \tag{21}$$

Lagrangian, therefore, reduces to

$$L(w, b, \alpha) = \sum_{i=1}^{N} \alpha_i - 0.5 \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j X_i X_j. \tag{22}$$

G. Ensemble Classification

Ensemble methods, combining results from different or similar classification models, were used extensively in the literature to improve the accuracy. Boosting and bagging are the most widely used and accepted ensemble methods. Ensembling also reduces the variability in classification [34]. While the primary motive of ensembling is reduction in the error rate, it also aids in reducing the variance of classifiers with high inherent variability such as decision tree. If the ensemble function comprises of N weak binary classifiers, each with an error rate of ε, binomial probability distribution can be used to evaluate the error rate of the ensemble. The ensemble will choose the output class based on the output of the majority classifiers; hence, an error will occur when the majority of classifiers make an error. For simplicity, we assume that the error classifications of the classifiers are uncorrelated. The probability of k out of N classifiers exhibiting an error is given by

$$\binom{N}{k} \varepsilon^k (1 - \varepsilon)^{n-k}. \tag{23}$$

Hence, for the ensemble to exhibit an error, the error rate is given by [34]

$$\varepsilon_{\text{ensemble}} = \sum_{i=\lceil (N+1)/2 \rceil}^{N} \binom{N}{k} \varepsilon^k (1 - \varepsilon)^{(n-i)}. \tag{24}$$

When $\varepsilon < 0.5$ and $k > 1$, $\varepsilon_{\text{ensemble}} \ll \varepsilon$.

Hence, for weak classifiers with classification error rates less than 0.5, it is reasonable to use an ensemble method. The prediction error can further be decomposed into the noise, variance, and bias. If a value x_i is predicted as y_i by the classifier, the error is given by

$$\varepsilon = x_i - y_i = \text{noise} + \text{variance} + \text{bias}. \tag{25}$$

Noise is the lowest bound that the classification can achieve, and variance gives the variability in the predicted values y_i. Bias on the other hand is the average distance between the predicted (y_i) and the actual values (x_i). While boosting reduces both the variance and the bias, bagging helps only in variance reduction.

V. Experiment Settings

A. Classifier Implementations and Parameters

1) Naïve Bayes: Classical probabilistic naïve Bayes classifier is used in our experiment. Supervised discretization is carried out for the numeric attributes. Normal distribution is assumed for the attributes, though kernel density estimator can be used as well. The classifier implementation uses Laplacian estimate to avoid the zero frequency problem. Underflow prevention in the case of extremely low posterior probabilities is resolved using the logarithmic scale for probabilities.

2) k-Nearest Neighbors: Our training method uses IBk [34] algorithm, which is an implementation of k-NN classifier. The algorithm uses classic Euclidean distance equation (5) as a metric. 1-NN is used in this set of experiments. Linear search is used for neighbor search. The choice of 1-NN is made due to the appealing theoretic property of the classifier that with an increase in sample size, it approaches the optimum error rate, defined quantitatively by the Bayes error rate. The Bayes error rate [35] specifies the minimum achievable error rate for a given data distribution. The 1-NN algorithm guarantees to achieve twice the Bayes error rate for very large data sets [32]. Mathematically, this can be represented as follows. As the number of samples in the training data set $N \rightarrow \infty$, $\varepsilon_{1NN} \leq 2\varepsilon_{\text{bayes}}$.

This also gives an intuitive way to calculate the lower bound on the Bayes error rate

$$\text{lowerBound}(\varepsilon_{\text{bayes}}) = \varepsilon_{(1NN)}/2. \tag{26}$$

This lower bound can thereafter be used to benchmark the performance of other classifiers. Optimum values of k can be obtained through cross validation to asymptotically achieve the Bayes error rate. Theoretically, as $k \to \infty$ and $N \to \infty, \varepsilon(k - nn) \to \varepsilon_{\text{bayes}}$.

3) Decision Tree: This paper uses J48 classifier, which is an implementation of the C4.5 decision tree. C4.5 [36] is an improvisation of the ID3 algorithm for tree generation. Entropy is used as a measure of information. C4.5 considers missing values and attributes with continuous range. A pruned tree with a pruning confidence (C_{prune}) of 0.25 is used in this paper. The pruning confidence defines the confidence threshold for the pruning tree [37]. It is a pessimistic upper bound on the error rate at a particular leaf. A smaller value indicates a higher bound and heavier pruning of the decision tree. The minimum number of instances per leaf node of the tree is chosen to be two.

4) Support Vector Machine: In this paper, LibSVM [38] implementation of SVM is used. Kernel type is set to the default choice of radial basis function. Linear kernel function is another alternative, but the choice of radial basis function is made due to better performance. A complexity value of 1.0 is used. The complexity value governs the tradeoff between the large margin and number of misclassified training points [39]. A large value of C indicates that the model attempts to classify all the training points correctly, thereby reducing the margin. A smaller C on the other hand would allow a few training points to be misclassified, but would make the model susceptible to noise by adequately enhancing the margin.

5) AdaBoost: Due to its effectiveness in both bias and variance reduction, boosting is chosen as a preferred ensemble method. AdaBoost [40], [41], which is an adaptive boosting algorithm, is used in this paper to attempt an increase in the prediction accuracy. The effectiveness of the algorithm in the enhancement of the prediction accuracy can be attributed to the fact that it is a probably approximately correct algorithm [32]. This implies that the returning hypothesis is consistent on training data set of large volume. Various works in the literature explain the reasons behind the efficiency of AdaBoost [42], [43]. The pseudocode for the algorithm is as follows [44].

Modeling Phase:

1) Assign equal weights to all instances $w_i = 1/N$, where N is the number of instances, $i = 1$ to N.
2) For $t = 1$ to M, apply classifier to weighted training set. Store generated hypothesis h_t.
3) Calculate error of the resulting hypothesis e_t.
4) If ($e_t == 0 || e_t \geq 0.5$), terminate modeling.
5) For all $i = 1$ to N, if ($\text{instance}_{\text{predicted}} == \text{instance}_{\text{actual}}$) then $w_{i+1} = w_i * e_t/(1 - e_t)$.
6) Normalize w for all instances.

Classification Phase:

1) Assign classwt = 0 for all classes.
2) For $t = 1$ to M, for each hypothesis, the following holds.
   a) Predict class weight using h_t, classwt_t.
   b) Add $-\log e/(1 - e)$ to the predicted weight.
   c) Return class (max weight).

B. Metrics Used

The description of the metrics used to characterize the classifier performance is as follows [31].

1) The correctly classified instances.
2) Incorrectly classified instances.
3) *Accuracy:* The correctly and incorrectly classified instances combined with the total number of instances are used to calculate the accuracy of the identifier.
4) Mean absolute error [31]

$$\text{MAE} = \left[\sum_{i=1}^{N} \mid y_i - x_i \mid\right] \Big/ N \tag{27}$$

where
- x_i actual class;
- y_i predicted class;
- N total number of instances in the training set;
- x_{mean} ensemble average of the sample set.

5) Root mean squared error

$$\text{RMSE} = \sqrt{\left[\sum_{i=1}^{N} (y_i - x_i)^2\right] \Big/ N}. \tag{28}$$

6) Relative absolute error

$$\text{RAE} = \left[\sum_{i=1}^{N} \mid y_i - x_i \mid\right] \Big/ \sum_{i=1}^{N} \mid x_{\text{mean}} - x_i \mid . \tag{29}$$

7) Root relative squared error

$$\text{RRSE} = \sqrt{\frac{\left[\sum_{i=1}^{N} (y_i - x_i)^2\right]}{\sum_{i=1}^{N} (x_i - x_{\text{mean}})^2}}. \tag{30}$$

8) Kappa statistic is a parametric measure of the agreement of the predicted value with the true class. A unity value of this parameter signifies complete agreement between the two. A positive value of this variable signifies that the classifier is performing better than what would have been predicted by chance.

Some other related metrics used to evaluate the performance of the classifier are as follows [32].

1) Sum of square error

$$\text{SSE} = \sum_{i=1}^{N} (x_i - y_i)^2. \tag{31}$$

2) Sum of square regression

$$\text{SSE} = \sum_{i=1}^{N} (y_i - y_{\text{mean}})^2. \tag{32}$$

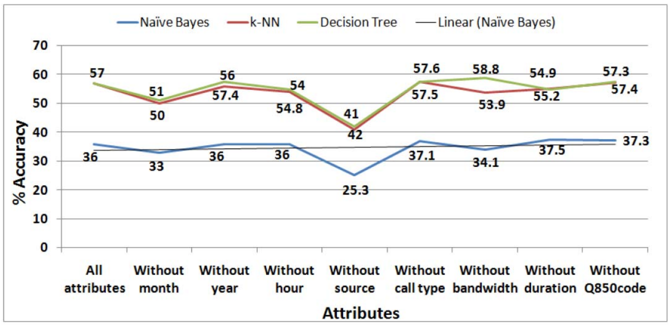


Fig. 8. Learning accuracy of the classifier by using various attribute sets for destination prediction.

3) Sum of square total

$$\mathrm{SST} = \sum_{i=1}^{N} (x_i - x_{\mathrm{mean}})^2. \tag{33}$$

4) Coefficient of determination

$$D^2 = \mathrm{SSR}/\mathrm{SST} \tag{34}$$

where $D = 1$ for a perfect model.

VI. Results and Discussion

The experiments are conducted with two oversights in mind. While one set of experiment is conducted to predict the destination of a VC given source and other attributes, the other is to predict the bandwidth of connection given the source, destination, and call-time parameters.

A. Destination Prediction

In the first set of experiments, the four classifiers are trained to predict the call destination if a prior knowledge of other attributes is assumed. Supervised learning is employed with tenfold cross validation [45]. Ten iterations of this process are carried out with a different set being used for testing every time. The average percentage accuracy yielded by various classifiers, with all the features combined as well as by removing one attribute every time is represented in the form of a graph in Fig. 8. SVM classifier exhibits worst performance in terms of % accuracy as well as the processing time; hence, the results are omitted. The degraded performance of SVM is in contradiction to the related results widely published in the literature that claim its superiority in comparison with other learning algorithms. A notable observation that deserves attention from the plot is that the point of maximum accuracy (58.8%) is the one that excludes bandwidth as an attribute. This can be attributed to the high degree of randomness in the bandwidth between a specified source–destination pair. The continuous nature to the attribute, duration, convolutes the learning. However, even its exclusion does not increase the accuracy as much as the bandwidth exclusion does. The summary of cross validation using decision tree on all the attributes is recorded in Table II.

TABLE II

Summary of Cross Validation Using Decision Tree on All Attributes

Correctly classified	13132, 57.5965%
Incorrectly classified	9668, 42.4035%
Hour	0-23
Kappa statistic	0.5409
Mean absolute error	0.0092
Root mean squared error	0.0749
Relative absolute error	54.7018%
Root relative squared error	81.5321%

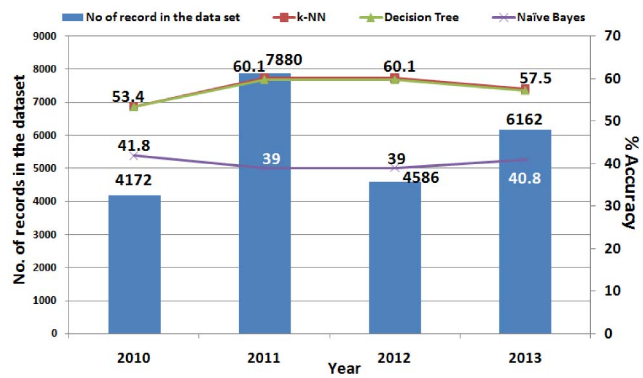


Fig. 9. Yearwise count of records and destination prediction accuracy with all attributes.

1) Singular Attribute Analysis: In the next iteration of experiments, prediction accuracy of destination is calculated with a singular attribute at a time, and the results are shown in Fig. 9. Partially in sync with the logically expected view, though source as a singular attribute can predict the destination only with average accuracy, when coupled with another attribute can predict with a fair accuracy. The prediction accuracy is 42.3% for learning with all the attributes except the source. The results seem unacceptable when observed in isolation. However, if the interpretation is carried out along with Fig. 8, the value of source as a predictor emerges. Source as a predictor can increase the prediction accuracy from 42.3% to 57%, which is a noteworthy increment.

2) Yearwise Corpus: A few peculiarities are observed on analyzing the yearwise breakup of the data set under consideration. Fig. 9 shows the details of the video call records per year. Conventionally, the learning accuracy increases with the increase in the corpus size. However, in the data set used for this paper, the learning model yields a greater accuracy of 60.1% with a data set of smaller size. Another deviation from the conventional trends is the increase in accuracy in 2012 than in 2013 despite being up to 25% smaller in sample size. The analysis of these deviations lead to is that the records in 2012 are in maximum order (minimum entropy) followed by the data set in 2011 and 2013. The randomness in the data of 2011 and 2013 is comparable as 2011 has a larger data set for enhancing its prediction accuracy by learning.

B. Bandwidth Prediction

In the second set of experiments, we attempt to train the four classifiers for prediction of bandwidth at which the VC

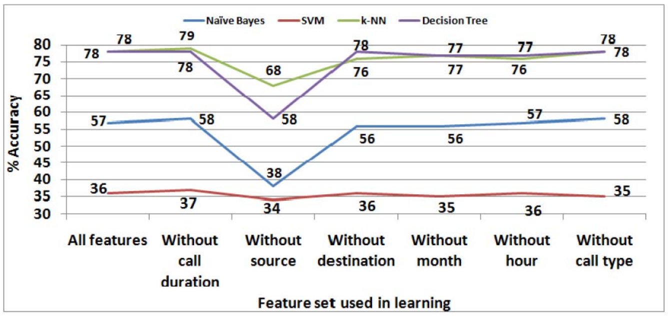


Fig. 10. Learning accuracy of the classifiers with varying features for bandwidth prediction.

a	b	c	d	e	f	g	h	i	Classified as
4238	622	91	357	122	12	0	0	0	a = 512
552	7425	62	385	174	43	0	0	0	b = 2048
98	151	621	75	21	17	0	0	0	c = 1536
348	605	43	3959	79	41	0	0	0	d = 1024
164	205	10	80	1623	211	0	0	0	e = 4096
6	7	29	18	105	1977	2	0	0	f = 6144
0	0	0	0	0	13	14	0	0	g = 3072
0	0	0	0	0	0	0	2	0	h = 3584
0	1	0	0	1	0	0	0	0	i = 2560

Fig. 11. Confusion matrix for bandwidth classifier using all feature set with bandwidth values in groups of 512 kb/s.

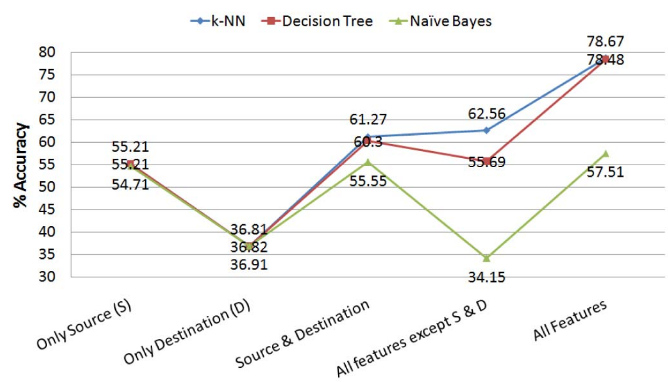


Fig. 12. % Accuracy of the classifier algorithms with specific feature set for bandwidth prediction.

call will be connected. Here, again cross validation is used for the performance evaluation over the hold-out validation method due to the finite size of the data set. While the hold-out validation method divides the data set into training and testing subsets, cross validation tests on each point as each set is used iteratively for both training and testing. The tenfold cross validation is used for the training and mean accuracy is calculated [46].

1) Prediction Accuracy With All Features: In the initial experiment subset, accuracy is calculated with all the predictors used for learning. The accuracy of 78.67% is achieved using k-NN, while decision tree yields an accuracy of 78.48% in predicting the bandwidth. Naïve Bayes and SVM achieved a lower accuracy of 57.51% and 36%, respectively. The reasons for SVM's lower accuracy, for this particular data set, require further research. A logical explanation for this degraded performance is the higher number classes and the heterogeneity of the attributes used. This result also indicates that a number of further investigations on SVM parameter optimization are required to fine-tune its operation with respect to the particular data set.

2) Accuracy With Reduced Feature Set: To investigate the effect of each individual feature in predicting the bandwidth, the features are removed one by one from the learning set, and the accuracy of predicting the bandwidth without a particular feature and using all the four learning algorithms is calculated, which is shown in Fig. 10 [47].

3) Variation in Class Segmentation: In the experiment subset, VCs are conducted at 30 different bandwidths. As reported in [45], the prediction accuracy increases with increase in the class difference. Hence, the bandwidths are segmented into groups of 512 kb/s, thereby consolidating the bandwidth values to nine distinct segments, i.e., 0–512, 512–1024, 1024–1536, and so on. With this increased class difference on the same data set and using all the features, k-NN, naïve Bayes, and decision tree yield an accuracy of 80.13%, 48.63%, and 80.69%, respectively. Fig. 11 shows the confusion matrix for decision tree classifier.

4) Analysis of Prediction Results: As can be analyzed from Fig. 10, the source is the main feature that helps in predicting the bandwidth, and hence, the independent prediction of bandwidth based on certain special combination of features is attempted, as shown in Fig. 12. As depicted by the accuracy values for all the trained classifiers in Fig. 12, it can be concluded that source and destination as a combined feature plays a decisive role in predicting the bandwidth. Addition of other features further augments the learning by increasing the accuracy significantly.

5) Accuracy With Ensemble: AdaBoost ensemble technique is employed on the classifiers in an attempt to increase the prediction accuracy. The accuracy of k-NN and naïve Bayes is 78.67% and 56.5%, respectively. The benefit of combining these classifiers is, therefore, miniscule. However, the performance of SVM significantly improves with a prediction accuracy of 78.6%. Boosting, therefore, contributes a commendable increase of 36% in the performance of the algorithm. A reason of this improved performance is the susceptibility of AdaBoost to outliers and noise. This observation also deviates from the regular mining trends that boosting helps improve the performance of classifiers with high variance. As SVM is a stable classifier with low variance, this result too needs further research and exploration.

C. Explanation

1) No Free Lunch Theorem: During the two sets of experiments for destination and bandwidth prediction, an unexpected observation was SVM exhibiting lower accuracy as compared with the other three classifiers. While this possesses an open research problem, an explanation of this result can be given by the no free lunch (NFL) theorem [48]. A simplistic description of the theorem is that no one algorithm or classifier is effective for all problem formulations. The classification problem can be represented as the formulation of a deterministic function $G(x)$ that takes a sample point, and predicts a class y_i, i.e., $x_i \epsilon X$, $y_i \epsilon Y$, and $y_i = G(x_i)$.

If C represents the classifier algorithm, the number of iterations is represented by $k = [1, 2, \ldots ||X|| - 1]]$. In the kth iteration, C accepts all input output combinations $TS_k = (x_1, y_1)(x_2, y_2), \ldots, (x_k, y_k)$ and outputs x_{k+1}, i.e., $C(TS_m) = x_{k+1}$. The NFL theorem [49] states that for classifiers $C1$ and $C2$

$$\sum_{G(x)} P(TS_k/G, k, T, C_1) = \sum_{G(x)} P(TS_k/G, k, T, C_2). \quad (35)$$

Equation (35) shows that the performance of the classifier at any iteration is independent of the classifying algorithm, if the evaluation is conducted over the entire function set. So heuristically, all classifiers have identical performance behavior when averaged over all functions.

a) Error estimation: The minimum error rate attainable on a given training data set is given by the Bayes error rate. The error defines the lower bound on the performance of a classifier for a given training data set. An ideal classifier approaches the Bayes error rate by selecting the class with the highest posterior probability. Mathematically, observation x classified into class C_k if $k = \arg\max P(C_i/x)$

$$\varepsilon_{\text{bayes}} = E[1 - \max(P(C_1/x)P(C_2/x)P(C_k/x)) \quad (36)$$

where E represents the expectation operator.

However, since the calculation of exact Bayes error rate is not feasible analytically, hence the Bhattacharya bound [50] is used to calculate the bounds on the error rate.

For a two class case, with normal distribution (means $\mu 1$ and $\mu 2$ and covariance matrices Σ_1 and Σ_2), bounds are given by

$$\begin{aligned} L_{\text{bhatt}} &= 1/8(\mu_1 - \mu_2)^T[0.5(\Sigma 1 + \Sigma 2)]^{-1}(\mu_1 - \mu_2) \\ &\quad + 0.5 \log \frac{|\, 0.5(\Sigma 1 + \Sigma 2)\,|}{\sqrt{\Sigma 1 \Sigma 2}}. \end{aligned} \quad (37)$$

VII. Application

The learning techniques presented in this paper can have several potential application areas. These can range from network analytics in social systems to command and control in defense. The need for prediction already exists in networks and was the motivation behind undertaking this paper. The aim was to strengthen the performance of underlying networks, which are an important component of the CPSS, which has been a revolutionizing concept these days. The idea is to integrate the physical elements with the computational and social elements. Networks form the backbone of these interactions; hence, there is a pressing need to predict the link parameter, thereby facilitating effortless interactions. Though this paper directly addresses destination and bandwidth predictions, future extensions may be extended to any network improvisation application. The technique can predict *a priori*, the probability of a VC connection between specific source–destination, and the associated Internet service provider may be collaboratively involved to ensure uninterrupted link operation during critical video calls. Intrusion detection systems may be modeled based on the deviations and anomalies observed by the system [51]. The prediction of general communication trends can be used for a myriad of forecasting applications in social and physical applications. Deviation in network profiling detected through meticulous network monitoring can be a critical input for C^4I systems. Learning-based group correlation in the network can greatly improve the visualization [52], [53] of communication hierarchies, dominant nodes, and connectivity hotspots, thereby assisting in decision making.

VIII. Conclusion

This experimental paper reports a maximum accuracy of 81% for bandwidth prediction and 60% for destination prediction in a VC connection. It was further established that the prediction accuracy was enhanced significantly with additional features added to the learning process. Two observations that cropped up during this work were lesser accuracy of SVM as compared with other classifiers and greater accuracy with smaller data set during the trial runs.

Future work in this direction includes attempting an increase in the prediction accuracy by using ensemble algorithms on heterogeneous classifiers [54] and using neural networks [55]. Cross validation of the results with other data sets is also planned.

References

[1] L. S. Jin, L. S. Wei, S. Lau, and E. Karuppiah, “Link quality prediction for multimedia streaming based on available bandwidth and latency,” in *Proc. IEEE 38th Conf. Local Comput. Netw. Workshops (LCN Workshops)*, Oct. 2013, pp. 1025–1032.

[2] P. Sarigiannidis, E. Karapistoli, and A. A. Economides, “Detecting Sybil attacks in wireless sensor networks using UWB ranging-based information,” *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7560–7572, 2015.

[3] T. Lindhorst, G. Lukas, E. Nett, and M. Mock, “Data-mining-based link failure detection for wireless mesh networks,” in *Proc. 29th IEEE Symp. Reliable Distrib. Syst.*, Oct./Nov. 2010, pp. 353–357.

[4] E. Ostlin, H.-J. Zepernick, and H. Suzuki, “Small-scale fading prediction using an artificial neural network,” in *Proc. IEEE 61st Veh. Technol. Conf.*, May/Jun. 2005, pp. 82–86.

[5] S. Goswami, S. Misra, and S. Jain, “Cognitive correlation of source-destination pair in a video conference network using call attributes,” in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, New Delhi, India, Dec. 2014, pp. 1–5.

[6] D. Horovitz and S. Dolev, “Collabrium: Active traffic pattern prediction for boosting P2P collaboration,” in *Proc. 18th IEEE Int. Workshops Enabling Technol., Infrastruct. Collaborative Enterprises*, Jun./Jul. 2009, pp. 116–121.

[7] A. Bashar, “Autonomic scaling of cloud computing resources using BN-based prediction models,” in *Proc. IEEE 2nd Int. Conf. Cloud Netw. (CloudNet)*, Nov. 2013, pp. 200–204.

[8] A. Ganapathi *et al.*, “Predicting multiple metrics for queries: Better decisions enabled by machine learning,” in *Proc. IEEE 25th Int. Conf. Data Eng. (ICDE)*, Washington, DC, USA, Mar./Apr. 2009, pp. 592–603.

[9] N. Williams, S. Zander, and G. Armitage, “Evaluating machine learning algorithms for automated network application identification,” Center Adv. Internet Archit., Swinburne Univ. Technol., Melbourne, VIC, Australia, Tech. Rep. 060410B, 2006, pp. 1–14.

[10] R. Beverly, K. Sollins, and A. Berger, “SVM learning of IP address structure for latency prediction,” in *Proc. SIGCOMM Workshop Mining Netw. Data (MineNet)*, Pisa, Italy, 2006, pp. 299–304.

[11] J. Littman and M. Boyan, *Advances in Neural Information Processing Systems*. San Mateo, CA, USA: Morgan Kaufmann, 1994.

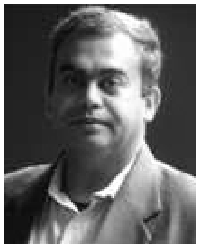
[12] N. Feamster and J. Rexford, “Network-wide prediction of BGP routes,” *IEEE/ACM Trans. Netw.*, vol. 15, no. 2, pp. 253–266, Apr. 2007.

[13] R.-H. Cheng and C. Cheng, "Efficient prediction-based location updating and destination searching mechanisms for geographic routing in mobile ad hoc networks," *J. Inf. Sci. Eng.*, vol. 28, no. 1, pp. 115–129, 2012.
[14] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "Sector-based routing with destination location prediction for underwater mobile networks," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, 2009, pp. 1148–1153.
[15] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian neural networks for Internet traffic classification," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 223–239, Jan. 2007.
[16] N. Ye, Y. Zhang, and C. M. Borror, "Robustness of the Markov-chain model for cyber-attack detection," *IEEE Trans. Rel.*, vol. 53, no. 1, pp. 116–123, Mar. 2004.
[17] W. Li and Q. Li, "Using naïve Bayes with AdaBoost to enhance network anomaly intrusion detection," in *Proc. 3rd Int. Conf. Intell. Netw. Intell. Syst.*, 2010, pp. 486–489.
[18] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, "Internet traffic classification by aggregating correlated naïve Bayes predictions," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 5–15, Jan. 2013.
[19] M. P. Singh, "Quantifying concept drifting in network traffic using ROC curves from naïve Bayes classifiers," in *Proc. Nirma Univ. Int. Conf. Eng.*, Nov. 2013, pp. 1–5.
[20] F. Gumus, C. O. K. Sakar, Z. Erdem, and O. Kursun, "Online naïve Bayes classification for network intrusion detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2014, pp. 670–674.
[21] R. L. Gomes and E. R. M. Madeira, "A traffic classification agent for virtual networks based on QoS classes," *IEEE Latin Amer. Trans.*, vol. 10, no. 3, pp. 1734–1741, Apr. 2012.
[22] P. R. Kalva, F. Enembreck, and A. L. Koerich, "WEB image classification using classifier combination," *IEEE Latin Amer. Trans.*, vol. 6, no. 7, pp. 661–671, Dec. 2008.
[23] J. F. C. Joseph, B.-S. Lee, A. Das, and B.-C. Seet, "Cross-layer detection of sinking behavior in wireless ad hoc networks using SVM and FDA," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 2, pp. 233–245, Mar./Apr. 2011.
[24] X. Jun, Y. Lu, Z. Lei, and X. Hui, "Boosting decision stumps to do pairwise classification," *Electron. Lett.*, vol. 50, no. 12, pp. 866–868, Jun. 2014.
[25] S. Pan, J. Wu, X. Zhu, and C. Zhang, "Graph ensemble boosting for imbalanced noisy graph stream classification," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 954–968, May 2015.
[26] J. Xu, Q. Wu, J. Zhang, F. Shen, and Z. Tang, "Boosting separability in semisupervised learning for object classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1197–1208, Jul. 2014.
[27] Y. Chi and F. Porikli, "Classification and boosting with multiple collaborative representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1519–1531, Aug. 2014.
[28] S. Goswami, S. Verma, and N. R. Krishnan, "Relationship visualization between books and users based on mining library circulation data," in *Proc. 1st Int. Conf. Intell. Interact. Technol. Multimedia (IITM)*, 2010, pp. 109–113.
[29] S. Sarkar, P. Mitra, and S. Goswami, "Benchmarking of text mining tools: Hague Centre for Strategic Studies," Hague Centre Strategic Studies, The Netherlands, Tech. Rep., 2010.
[30] D. S. Sayad. (Jun. 2010). *An Introduction to Data Mining*. [Online]. Available: http://www.saedsayad.com
[31] O. Okun, *Feature Selection and Ensemble Methods for Bioinformatics: Algorithmic Classification and Implementations*. Hershey, PA, USA: IGI Global, 2011.
[32] R. Khanna and M. Awad, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. New York, NY, USA: Apress, 2015.
[33] D. T. Larose and C. D. Larose, *Data Mining and Predictive Analytics*, 2nd ed. Hoboken, NJ, USA: Wiley, 2015.
[34] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2000.
[35] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann, 1993.
[36] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
[37] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.
[38] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, *A Practical Guide to Support Vector Classification*. Taipei, Taiwan: National Taiwan Univ., 2010.
[39] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, Bari, Italy, 1996, pp. 148–156.
[40] P. Ramzi, F. Samadzadegan, and P. Reinartz, "Classification of hyperspectral data using an AdaBoost SVM technique applied on band clusters," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 7, no. 6, pp. 2066–2079, Jun. 2014.
[41] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.
[42] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Pearson Education, 2003.
[43] I. H. Witten, E. Frank, M. A. Hall, and G. Holmes, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2011.
[44] S. Goswami, S. Kapoor, and P. Bhardwaj, "Machine learning for automated tender classification," in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2011, pp. 1–4.
[45] S. Misra and S. Goswami, "Cognitive prediction of end-to-end bandwidth utilisation in a non-QoS video conference," in *Proc. IEEE Int. Conf. Commun.*, London, U.K., Jun. 2015, pp. 6152–6156.
[46] C. Manasseh and R. Sengupta, "Predicting driver destination using machine learning techniques," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 142–147.
[47] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[48] A. Y. S. Lam and V. O. K. Li, "Generalization of the no-free-lunch theorem," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2009, pp. 4322–4328.
[49] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, no. 1, pp. 99–109, 1943.
[50] A. Azaria, A. Richardson, S. Kraus, and V. S. Subrahmanian, "Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data," *IEEE Trans. Comput. Social Syst.*, vol. 1, no. 2, pp. 135–155, Jun. 2014.
[51] J. Leskovec and E. Horvitz, "Geospatial structure of a planetary-scale social network," *IEEE Trans. Comput. Social Syst.*, vol. 1, no. 3, pp. 156–163, Sep. 2014.
[52] E. Karapistoli, P. Sarigiannidis, and A. A. Economides, "SRNET: A real-time, cross-based anomaly detection and visualization system for wireless sensor networks," in *Proc. 10th Workshop Vis. Cyber Secur.*, 2013, pp. 49–56.
[53] P. Sarigiannidis, E. Karapistoli, and A. A. Economides, "VisIoT: A threat visualisation tool for IoT systems security," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 2633–2638.
[54] A. Eswaradass, X.-H. Sun, and M. Wu, "A neural network based predictive mechanism for available bandwidth," in *Proc. 19th IEEE Int. Conf. Parallel Distrib. Process. Symp.*, Apr. 2005, p. 33a.
[55] S. Goswami and M. S. Shishodia, "A fuzzy based approach to stylometric analysis of blogger's age and gender," in *Proc. 12th Int. Conf. Hybrid Intell. Syst. (HIS)*, 2012, pp. 47–51.

Sudip Misra (SM'11) received the Ph.D. degree in computer science from Carleton University, Ottawa, ON, Canada.

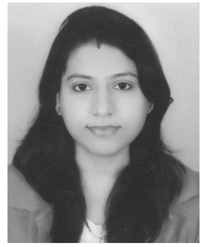
He was associated with Cornell University, Ithaca, NY, USA, Yale University, New Haven, CT, USA, Nortel Networks, Mississauga, ON, Canada, and the Government of Ontario in Canada. He is currently an Associate Professor with the Department of Computer Science Engineering, IIT Kharagpur, Kharagpur, India.

Dr. Misra has won eight research paper awards in different conferences. He received the IEEE ComSoc Asia Pacific Outstanding Young Researcher Award at the IEEE GLOBECOM 2012. He also received the Canadian Governments Prestigious NSERC Post-Doctoral Fellowship and the Humboldt Research Fellowship in Germany.



Sumit Goswami received the B.Tech. degree in computer science and engineering from Punjab Technical University, Jalandhar, India, the M.Tech. degree in computer science and engineering from IIT, Kharagpur, Kharagpur, India, and the Post Graduate Diploma degree in journalism and mass communication from Indira Gandhi National Open University, Delhi. He is currently pursuing the Ph.D. degree from IIT Kharagpur.

He is with the Defence Research and Development Organization, New Delhi, India. He has authored about 60 papers/chapters in various journals, books, data competitions, and conferences.



Chaynika Taneja received the B.Tech. degree in electronics and communication from the National Institute of Technology, Kurukshetra, India. She is currently pursuing the M.Tech. degree in electronics and communication engineering with IIT Delhi, New Delhi, India.

She is a Scientist with the Defence Research and Development Organization, New Delhi. She has authored technical papers for international conferences, journals, and magazines. Her current research interests include advanced approaches to network management, mobile agents, and secured communication.