A FRAMEWORK FOR DISTRIBUTED MULTIMEDIA STREAM MINING SYSTEMS USING COALITION-BASED FORESIGHTED STRATEGIES

Hyunggon Park, Deepak S. Turaga+, Olivier Verscheure+ and Mihaela van der Schaar**

+IBM T. J. Watson Research Center, Hawthorne, NY, USA
*UCLA Electrical Engineering Department, Los Angeles, CA, USA

ABSTRACT

In this paper, we propose a distributed solution to the problem of configuring classifier trees in distributed stream mining systems. The configuration involves selecting appropriate false-alarm detection tradeoffs for each classifier to minimize end-to-end penalty in terms of misclassification cost. In the proposed solution, individual classifiers select their operating points (i.e., actions) to maximize a local utility function. The utility may be purely local to the current classifier, corresponding to a *myopic* strategy, or may include the impact of the classifier actions on successive classifiers in the tree, corresponding to a *foresighted* strategy. We analytically show that actions determined by the foresighted strategies can improve the end-to-end performance of the classifier tree and derive an associated probability bound. We then evaluate our solutions on an application for hierarchical sports scene classification. By comparing centralized, myopic and foresighted solutions, we show that foresighted strategies result in better performance than myopic strategies, and also asymptotically approach the centralized optimal solution.

Index Terms— Resource constrained stream mining, coalition-based foresighted strategy, binary classifier tree.

1. INTRODUCTION

Emerging applications, such as online photo and video streaming services, financial analysis, real-time manufacturing process control, search engines, spam filters, security, and medical services [1, 2] require processing and classification of continuous, high volume data streams. Scaling and resource considerations lead to such applications being developed as processing topologies of distributed operators [3, 4] deployed on large-scale stream mining systems [4,5]. Several of these stream mining applications implement topologies (ensembles such as trees or cascades) of low-complexity binary classifiers to hierarchically filter the data streams and jointly accomplish the task of complex classification [2, 6].

A key challenge for such applications involves management of individual classifiers to maximize end-to-end performance – especially under dynamically varying and distributed resource constraints and data characteristics. In this paper, we focus on the classifier configuration problem for binary tree topologies, i.e., determining the optimal operating point (detection - false alarm tradeoff) for each classifier in the tree, in order to maximize the end-to-end classification performance. Previously, this problem has been modeled as as an optimization problem and centralized techniques such as Sequential Quadratic Programming (SQP) [7] have been used to solve it. However, a centralized solution suffers disadvantages in terms of having a single central point of control and associated failure, issues with scaling and adaptation as the topology grows, and not allowing large scale applications with capabilities distributed across multiple proprietary entities.

In this paper, we propose a distributed solution, where each classifier decides its optimal action – selecting an operating point – in order to maximize a local utility function. Different optimal actions may be determined based on the availability of information about other classifiers. If only local information is available, the optimal action is *myopically* selected to maximize the classifier's own utility [8]. However, if a classifier has additional information about its successive classifiers, it can form a *coalition* with them and determine a *foresighted* action to maximize a coalition utility. We analytically show that foresighted actions improve the end-to-end performance of the classifier tree and derive an associated probability bound. Simulation results show that foresighted actions result in better performance than myopic actions. Moreover, we show that the foresighted actions approach the centralized optimal solution, as the coalition size and the number of actions increase.

This paper is organized as follows. In Section 2, we introduce our model for individual classifiers and classifier trees. In Section 3, we propose coalition-based foresighted strategies for classifier tree configuration. We present simulation results in Section 4, and conclude in Section 5.

2. DISTRIBUTED BINARY CLASSIFIER TREES

Consider a stream mining application [7], which consists of several binary classifiers in a tree topology depicted in Fig. 1. The topology of classifiers in this example is used to identify semantic concepts from sports image data using hierarchical filtering. Leaf classifiers (e.g. classifier 4, 8 etc.) represent the actual class of interest, while intermediate classifiers assist in hierarchical filtering of data based on a semantic hierarchy of concepts. The following descriptions can also be found in [8].

· *Configuration of Binary Classifier:* A binary classifier

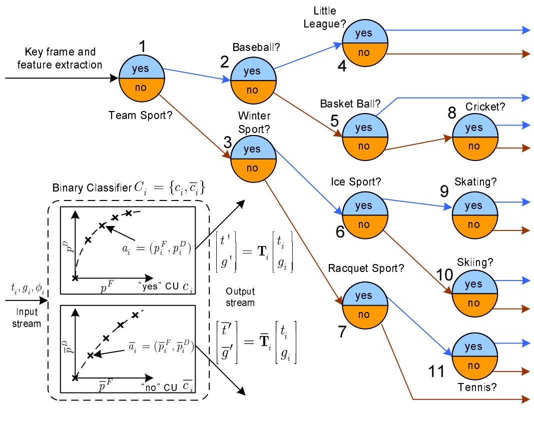

Fig. 1. An illustrative classifier topology.

filters input data into the "yes" class and the "no" class. We model each classifier C_i with two classification units (CUs), i.e., $C_i = \{c_i, \bar{c}_i\}$, corresponding to the "yes" and "no" outputs respectively (Fig. 1). We use notation $i \rightsquigarrow k$ to denote that c_i (or $\bar{c}_i$) is a *preceding* CU for c_k (or $\bar{c}_k$), or c_k (or $\bar{c}_k$) is *successive* CU for c_i (or $\bar{c}_i$). The topology allows disambiguation between the two CUs per classifier.

· *Stream Characteristics:* The input stream for classifier C_i is characterized by *throughput* t_i and *goodput* g_i, which represent total data rate and correctly labeled data rate, respectively. The true fraction of positive stream data for CU c_i is denoted by ϕ_i and is pre-determined, based on the classifier topology and data characteristics. For $\bar{c}_i$, $\bar{\phi}_i = 1 - \phi_i$.

· *Performance of CU:* As in [7], performance of c_i ($\bar{c}_i$) is controlled by its tradeoff between probability of false alarm p_i^F ($\bar{p}_i^F$) and probability of detection p_i^D ($\bar{p}_i^D$). The two CUs may have decoupled operating points, e.g. through the use of independent thresholds (one for "yes" and one for "no") for score based classifiers. The set of operating points (p_i^F, p_i^D) represent the quantized DET curve[1] – a non-decreasing concave function. We define an *action set* $\mathbf{A}_i$, with action $a_i \in \mathbf{A}_i$ representing the selection of operating point (p_i^F, p_i^D) (see Fig. 1). We can similarly define action set $\bar{\mathbf{A}}_i$ for CU $\bar{c}_i$.

· *Misclassification Cost:* Cost coefficients λ_i^F ($\bar{\lambda}_i^F$) and λ_i^M ($\bar{\lambda}_i^M$) represent the cost/penalty per unit data rate of *false alarm* and *miss* for CU c_i ($\bar{c}_i$). These coefficients are specified by the application for leaf classifiers – and may be derived for other classifiers based on the topology. λ_i^F and λ_i^M for intermediate CU c_i may be derived from its immediately successive classifier C_k as $\lambda_i^F = \phi_k \lambda_k^F + \bar{\phi}_k \bar{\lambda}_k^F$ and $\lambda_i^M = \phi_k \lambda_k^M + \bar{\phi}_k \bar{\lambda}_k^M$.

· *Input and Output Rates:* For c_i and $\bar{c}_i$, the output stream rates (t_i', g_i') and ($\bar{t}_i'$, $\bar{g}_i'$) may be derived as [7]:

$$\begin{bmatrix} t_i' \\ g_i' \end{bmatrix} = \mathbf{T}_i \begin{bmatrix} t_i \\ g_i \end{bmatrix}, \text{and} \begin{bmatrix} \bar{t}_i' \\ \bar{g}_i' \end{bmatrix} = \bar{\mathbf{T}}_i \begin{bmatrix} t_i \\ g_i \end{bmatrix}, \quad (1)$$

where $\mathbf{T}_i$ and $\bar{\mathbf{T}}_i$ are given by

[1]It can be referred to as Receiver Operating Characteristic (ROC) curve.

$$\mathbf{T}_i = \begin{bmatrix} p_i^F & \phi_i(p_i^D - p_i^F) \\ 0 & \phi_i p_i^D \end{bmatrix}, \text{and } \bar{\mathbf{T}}_i = \begin{bmatrix} \bar{p}_i^D & \phi_i(\bar{p}_i^F - \bar{p}_i^D) \\ 0 & \bar{\phi}_i \bar{p}_i^D, \end{bmatrix}.$$

· *Local Utility Function:* For CU c_i, the incurred cost due to misclassification is defined as $(t_i' - g_i')\lambda_i^F + (\Lambda_i - g_i')\,\lambda_i^M$. Λ_i represents a true fraction of stream data that belongs to c_i for input stream rate t_r to the tree and is defined as $\Lambda_i = t_r \phi_i \cdot \prod_{\forall k \in \{j | j \rightsquigarrow i\}} \hat{\phi}_k$, with $\hat{\phi}_k = \phi_k$ for c_k and $\hat{\phi}_k = \bar{\phi}_k$ for $\bar{c}_k$. We assume that Λ_i is known to c_i. The utility is then defined as the negative cost, or

$$U_i = -\left[(t_i' - g_i')\lambda_i^F + (\Lambda_i - g_i')\,\lambda_i^M\right]. \quad (2)$$

Similarly, we can define $\bar{U}_l$ for $\bar{c}_l$ in terms of $\bar{\lambda}_l^F$ and $\bar{\lambda}_l^M$. The end-to-end utility $U_{\mathbf{S}}$ may thus be expressed as

$$U_{\mathbf{S}} = \sum\nolimits_{c_l \in \mathbf{C}_L} U_l + \sum\nolimits_{\bar{c}_l \in \mathbf{C}_L} \bar{U}_l, \quad (3)$$

where $\mathbf{C}_L$ denotes a set of leaf CUs.

3. COALITION-BASED FORESIGHTED STRATEGIES FOR CLASSIFIER TREES

In this section, we study the impact of coalition formation and foresighted actions on the end-to-end application performance.

3.1. Available Information for CUs

The local information I_i required for CU c_i to define its utility in (2) is $I_i = \{t_i, g_i, \phi_i, \lambda_i^F, \lambda_i^M, \mathbf{A}_i, \Lambda_i\}$. Additionally, c_i can also get information $\mathbf{I}_{-i}$ about its *successive* classifiers. We have:

$$\mathbf{I}_{-i} = \{\hat{I}_k | i \rightsquigarrow k\}, \quad (4)$$

where $\hat{I}_k = I_k$ for c_k and $\hat{I}_k = \bar{I}_k$ for $\bar{c}_k$. Note that $\mathbf{I}_{-i}$ does not include I_i. $\bar{\mathbf{I}}_{-i}$ for $\bar{c}_i$ can be similarly defined. Information $\mathbf{I}_{-i}$ enables c_i to have a foresighted strategy.

3.2. Foresighted Strategy π_i of CU c_i

A strategy for CU c_i is to select an action that maximizes its utility. If c_i has information $\mathbf{I}_{-i}$ in addition to its local information I_i, it can select its action that maximizes the utility for the *coalition*. A coalition $\mathbf{G}_i$ of c_i is defined as the set of successive classifiers for which information is available to c_i. Such action selection strategy is referred to as *foresighted*. Specifically, the foresighted strategy of c_i with information $\mathbf{I}_i = \{I_i, \mathbf{I}_{-i}\}$ is denoted by $\pi_i(\mathbf{I}_i)$ and a foresighted action is selected as

$$a_i^* = \pi_i(\mathbf{I}_i) = \arg\max_{a_i \in \mathbf{A}_i} U_{\mathbf{G}_i}(a_i, (a_k, \bar{a}_k)_{C_k \in \mathbf{G}_i}), \quad (5)$$

where $U_{\mathbf{G}_i}$ denotes the utility achieved by $\mathbf{G}_i$, i.e., the sum of utilities derived by c_i's successive CUs with the largest distances in each branch. For example, in Fig. 2 (b), $U_{\mathbf{G}_f} = \hat{U}_f + \hat{\hat{U}}_f$. As a special case, if $\mathbf{I}_i = \{I_i\}$ (i.e., only local information is available), the strategy in (5) becomes *myopic*, since c_i selects its actions that maximize its own utility.

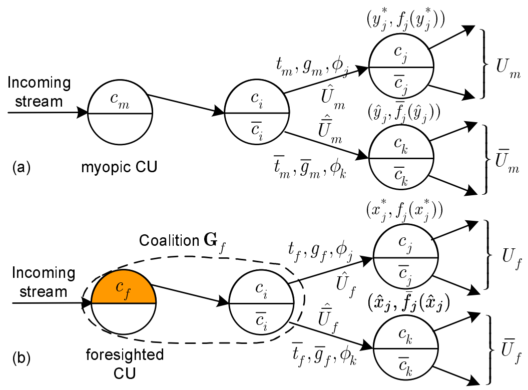

Fig. 2. Sub-trees without (a) and with (b) coalition.

Note that $U_{\mathbf{G}_i}$ depends on the action a_i of c_i as well as the actions of coalition members (i.e., $(a_k, \bar{a}_k)_{C_k \in \mathbf{G}_i}$). However, c_i decides its action a_i independently, based on the assumption that $(a_k, \bar{a}_k)_{C_k \in \mathbf{G}_i}$ are myopically determined. This keeps the foresighted solution completely distributed and allows the action a_i to be uniquely determined. However, compare to the myopic strategy, the computational complexity for the foresighted strategy increases. Note that a foresighted action of a CU only *indirectly* controls the action selections of successive classifiers.

3.3. Coalition and End-to-End Performance Improvement

In this section, we analytically show that a foresighted strategy can improve the end-to-end performance. In the following analysis, we consider an elementary sub-tree shown in Fig. 2. In this example, the end-to-end utilities of this tree are determined by $U_m + \bar{U}_m$ (a) and $U_f + \bar{U}_f$ (b). In Fig. 2(a), c_m determines its action based on a myopic strategy, while in Fig. 2(b), c_f decides its action based on a foresighted strategy for its coalition $\mathbf{G}_f = \{c_f, C_i\} = \{c_f, c_i, \bar{c}_i\}$.

A foresighted action by c_f guarantees that the coalition utility $\hat{U}_f + \hat{\bar{U}}_f \geq \hat{U}_m + \hat{\bar{U}}_m$, because the myopic strategy is a special case of the foresighted strategy. This may be decomposed as

$$\hat{U}_f - \hat{U}_m \geq \Delta\hat{u}, \text{ and } \hat{\bar{U}}_f - \hat{\bar{U}}_m \geq -\Delta\hat{\bar{u}}, \tag{6}$$

where $\Delta\hat{u} - \Delta\hat{\bar{u}} \geq 0$ for $\Delta\hat{u} \geq 0$. Based on the input-output relationship given in (1), the conditions in (6) can be expressed as

$$\Delta t/\Delta g \leq (\lambda_i^F + \lambda_i^M)/\lambda_i^F - \Delta\hat{u}/(\lambda_i^F \Delta g) \triangleq R_A, \tag{7}$$

$$\Delta\bar{t}/\Delta\bar{g} \leq (\bar{\lambda}_i^F + \bar{\lambda}_i^M)/\bar{\lambda}_i^F + \Delta\hat{\bar{u}}/(\bar{\lambda}_i^F \Delta\bar{g}) \triangleq R_B, \tag{8}$$

where $\Delta t \triangleq t_f - t_m$, $\Delta g \triangleq g_f - g_m$, $\Delta\bar{t} \triangleq \bar{t}_f - \bar{t}_m$, and $\Delta\bar{g} \triangleq \bar{g}_f - \bar{g}_m$. Without loss of generality[2], we assume that $\Delta g > 0$ and $\Delta\bar{g} > 0$.

[2]For $\Delta g < 0$ (or $\Delta\bar{g} < 0$), it only switches between R_C and R_A (or R_D and R_B) in (13), and thus, it does not affect the corresponding conclusions.

Now, we derive conditions on when the foresighted strategy of c_f leads to better end-to-end utility, i.e., when we have $U_f + \bar{U}_f > U_m + \bar{U}_m$. Consider the contradictory case, i.e., we have $U_m + \bar{U}_m \geq U_f + \bar{U}_f$. For this to happen, we have:

$$U_m - U_f \geq \Delta u, \text{ and } \bar{U}_m - \bar{U}_f \geq -\Delta u. \tag{9}$$

Using (1) and the fact that action y_j^* (or $\hat{y}_j$) instead of x_j^* (or $\hat{x}_j$) incurs higher costs for c_j (or $\bar{c}_j$) in Fig. 2, conditions in (9) can be expressed as

$$\Delta t/\Delta g \geq Q_j^N/Q_j^D + \Delta u/(\Delta g Q_j^D) \triangleq R_C, \tag{10}$$

$$\Delta\bar{t}/\Delta\bar{g} \geq Q_k^N/Q_k^D - \Delta u/(\Delta\bar{g} Q_k^D) \triangleq R_D, \tag{11}$$

where $Q_h^N = y_h^* \phi_h \lambda_h^F + f_h(y_h^*) \phi_h \lambda_h^M + \bar{f}_h(\hat{y}_h)(\bar{\lambda}_h^F + (1 - \phi_h)\bar{\lambda}_h^M) - \phi_h \hat{y}_h \bar{\lambda}_h^F$ and $Q_h^D = y_h^* \lambda_h^F + \bar{f}_h(\hat{y}_h)\bar{\lambda}_h^F$ for $h = \{j, k\}$. Based on this, we can conclude that

$$U_m + \bar{U}_m \geq U_f + \bar{U}_f \Rightarrow$$
$$R_C \leq \Delta t/\Delta g \leq R_A, \text{ and } R_D \leq \Delta\bar{t}/\Delta\bar{g} \leq R_B. \tag{12}$$

This finally leads to a lower bound for the probability that $U_m + \bar{U}_m < U_f + \bar{U}_f$, i.e.,

$$\Pr(U_m + \bar{U}_m < U_f + \bar{U}_f) \geq$$
$$1 - \Pr(R_C \leq \frac{\Delta t}{\Delta g} \leq R_A) \cdot \Pr(R_D \leq \frac{\Delta\bar{t}}{\Delta\bar{g}} \leq R_B). \tag{13}$$

In general, as the difference in coalition utility achieved by the foresighted and myopic strategies increases, i.e., $\Delta\hat{u}$ becomes large or $\Delta\hat{\bar{u}}$ becomes small (thus, R_A or R_B have smaller values), the lower bound for $\Pr(U_m + \bar{U}_m < U_f + \bar{U}_f)$ increases. Hence, if a foresighted strategy leads to significant improvements in local coalition utility, it is also more likely to increase the end-to-end utility.

4. SIMULATION RESULTS

4.1. Simulation Set-up

We consider the semantic concept detection application [7] shown in Fig. 1. Each classifier operates on low level image features such as color histograms, color correlograms, etc. using a Support Vector Machine, and classifiers are organized into a semantic hierarchy of concepts. We compare the performance of myopic and foresighted strategies against a centralized approach [7].

4.2. Impact of Action Set Size and Foresighted Strategy

To highlight the impact of the number of available actions and the foresighted strategies on the end-to-end application cost, we consider an elementary sub-tree of our application, consisting of classifiers 1, 2, and 3 in Fig. 1. We set $\lambda_l^F(\bar{\lambda}_l^F) = \lambda_l^M(\bar{\lambda}_l^M) = 1$, for $l = 2, 3$, and derive $\lambda_1^F(\bar{\lambda}_1^F)$ and $\lambda_1^M(\bar{\lambda}_1^M)$ appropriately. We set the input stream rate $t_r = 1$. The resulting misclassification costs for different number of actions

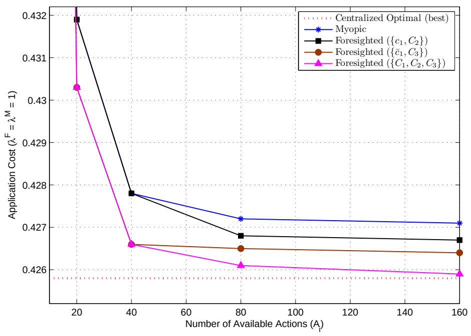

Fig. 3. End-to-end application costs.

($A_i = 20, 40, 80, 160$ for $i = 1, 2, 3$) and different degrees of foresightedness are shown in Fig. 3.

Fig. 3 clearly shows that increasing the number of available actions leads to a lower application cost – approaching the performance of the best result for the centralized algorithm. Moreover, this result shows that foresighted strategies always outperform myopic strategies. This is because any action part of the myopic strategy is always included in the candidate actions for the foresighted strategies. This result also shows that increasing the coalition size can result in a lower end-to-end application cost. Specifically, coalition $\{C_1, C_2, C_3\}$ achieves much lower cost than coalitions $\{c_1, C_2\}$ and $\{\bar{c}_1, C_3\}$, both of which outperform the myopic strategy. We also observe that the achieved utilities from the same size coalitions, $\{c_1, C_2\}$ and $\{\bar{c}_1, C_3\}$, are different. This is because the derived utilities depend on not only the foresighted actions (a_1^* and $\bar{a}_1^*$) but also their DET curves and $\hat{\phi}_l$, $l = 2, 3$. Hence, forming coalition with CUs having better DET performance or higher $\hat{\phi}_l$ can result in higher coalition utility.

4.3. End-to-End Application Performance

In this simulation, we assume that each CU has 80 available actions, and we set $\lambda_l^F(\bar{\lambda}_l^F) = \lambda_l^M(\bar{\lambda}_l^M) = 1$ for leaf CUs $c_l \in \mathbf{C}_L$. We consider three different foresighted strategies $\{c_1, C_2\}$, $\{\bar{c}_1, C_3\}$, and $\{C_1, C_2, C_3\}$, as in section 4.2. We compare against the centralized solution in [7]. Since SQP is gradient descent based, we use 500 different randomized starting points, and provide the minimum (Best) as well as the average cost (Average). The resulting end-to-end application costs are shown in Table 1.

It is clear that the proposed distributed approaches always outperform the average performance of the centralized approach. This is reflected in the percentages in the table, computed as $(Cost - Cost_{avg}^{cent})/(Cost_{best}^{cent} - Cost_{avg}^{cent}) \times 100\%$. As discussed in Section 3.3 and Section 4.2, enlarging a coalition size results in improved application performance. In the illustrative coalitions, while foresighted strategies for coalitions $\{c_1, C_2\}$ or $\{\bar{c}_1, C_3\}$ achieve approximately 70% of centralized best solution, foresighted decisions for coalition $\{C_1, C_2, C_3\}$ achieve 72.5% of centralized best solution.

Table 1. Achieved End-to-End Application Costs (80 actions)

Experiment Cases	End-to-End Application Cost
Centralized (Average)	0.5874 (0 %)
Myopic	0.5361 (68.5 %)
Foresighted (c_1, C_2)	0.5348 (70.2 %)
Foresighted ($\bar{c}_1, C_3$)	0.5345 (70.6 %)
Foresighted (C_1, C_2, C_3)	0.5331 (72.5 %)
Centralized (Best)	0.5125 (100 %)

5. CONCLUSIONS

In this paper, we propose a distributed solution to the configuration of classifier tree topologies in distributed stream mining system. Individual classifiers select actions (i.e., determine operating points) in a distributed way, based on myopic or foresighted strategies, depending on the available information. We show analytically that the foresighted strategies that maximize (local) coalition utilities can eventually improve the end-to-end application utility. Our simulation results, performed on a semantic concept detection application for sports image analysis, show that deploying foresighted strategies improve performance of the classifier tree application. Moreover, the performance incrementally improves as the coalition size increases. We also show that the proposed distributed approaches performance improves with the number of actions available to each classifier – asymptotically approaching the best performance of a centralized approach.

6. REFERENCES

[1] M. A. Shah, J. M. Hellerstein, S. Chandrasekaran, and M. J. Franklin, "Flux: An adaptive partitioning operator for continuous query systems," in *ICDE*, Mar. 2003, pp. 25–36.

[2] R. Lienhart, L. Liang, and A. Kuranov, "A detector tree for boosted classifiers for real-time object detection and tracking," in *ICME*, 2003, pp. 277–230.

[3] L. Amini, H. Andrade, F. Eskesen, R. King, Y. Park, P. Selo, and C. Venkatramani, "The stream processing core," IBM T.J. Watson Research Center, Tech. Rep. RSC 23798, Nov. 2005.

[4] M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Çetintemel, Y. Xing, and S. B. Zdonik, "Scalable distributed stream processing," in *CIDR*, Jan. 2003.

[5] M. Balazinska, H. Balakrishnan, S. Madden, and M. Stonebraker, "Fault tolerance in the borealis distributed stream processing system," in *SIGMOD*, Jun. 2005, pp. 13–24.

[6] Y. Mao, X. Zhou, D. Pi, Y. Sun, and S. T. C. Wong, "Multiclass cancer classification by using fuzzy support vector machine and binary decision tree with gene selection." *Journal of Biomedical Biotechnology*, vol. 2005, no. 2, pp. 160–71, 2005.

[7] D. S. Turaga, B. Foo, O. Verscheure, and M. van der Schaar, "Configuring topologies of distributed semantic concept classifiers for continuous multimedia stream processing," in *ACM Multimedia*, 2008, pp. 289–298.

[8] H. Park, D. S. Turaga, O. Verscheure, and M. van der Schaar, "Tree configuration game modeling for distributed multimedia stream mining systems," in *ICASSP '09*, Apr. 2009.