

Collaborative Learning Agents (CLA) for Swarm Intelligence and Applications to Health Monitoring of System of Systems

Ying Zhao¹[0000-0001-8350-4033] and Charles C. Zhou²[0000-0001-9598-015X]

¹ Naval Postgraduate School, Monterey, CA 93943, USA
yzhao@nps.edu

² Quantum Intelligence, Inc., Monterey, CA 93943, USA
charles.zhou@quantumii.com
<http://www.quantumii.com>

Abstract. The system of systems is the perspective of multiple systems as part of a larger, more complex system. A system of systems usually includes highly interacting, interrelated and interdependent sub-systems that form a complex and unified system. Maintaining the health of such a system of systems requires constant collection and analysis of the big data from sensors installed in the sub-systems. The statistical significance for machine learning (ML) and artificial intelligence (AI) applications improves purely due to the increasing big data size. This positive impact can be a great advantage. However, other challenges arise for processing and learning from big data. Traditional data sciences, ML and AI used in small- or moderate-sized analysis typically require tight coupling of the computations, where such an algorithm often executes in a single machine or job and reads all the data at once. Making a generic case of parallel and distributed computing for a ML/AI algorithm using big data proves a difficult task. In this paper, we described a novel infrastructure, namely collaborative learning agents (CLA) and the application in an operational environment, namely swarm intelligence, where a swarm agent is implemented using a CLA. This infrastructure enables a collection of swarms working together for fusing heterogeneous big data sources in a parallel and distributed fashion as if they are as in a single agent. The infrastructure is especially feasible for analyzing data from internet of things (IoT) or broadly defined system of systems to maintain its well-being or health. As a use case, we described a data set from the Hack the Machine event, where data sciences and ML/AI work together to better understand Navy's engines, ships and system of systems. The sensors installed in a distributed environment collect heterogeneous big data. We show how CLA and swarm intelligence used to analyze data from system of systems and quickly examine the health and maintenance issues across multiple sensors. The methodology can be applied to a wide range of system of systems that leverage collaborative, distributed learning agents and AI for automation.

Keywords: collaborative learning agents · swarm intelligence · system of systems · health monitoring · lexical link analysis · distributed com-

puting · parallel computing · data fusion · machine learning · artificial intelligence

1 Introduction

The system of systems is the viewing of multiple systems as part of a larger, more complex system. For example, a Navy ship is a system of systems. The internet of things (IoT) is a system of systems. A system of systems usually includes highly interacting, interrelated and interdependent sub-systems that form a complex and unified system. Maintaining the health of such a system of systems requires constant collection and analysis of the big data from sensors. The data for a system of systems are often collected in a distributed fashion from the sensors installed in the sub-systems. Fusing and analyzing the data from heterogeneous sensors in a holistic approach the key to successfully detecting problems, monitoring and maintaining the health of a system of systems.

In a separate perspective, as the size of data increases for data analytics such as machine learning (ML) and artificial intelligence (AI), the statistical significance for these methods often improves purely due to the increased data size. This positive impact of big data drives proliferated considerations of ML/AI applications.

However, other challenges arise. For example, the computational concept map/reduce - an analytic programming paradigm for big data, which consists of two tasks: 1) the “map” task, where an input data set is converted into key/value pairs; and 2) the “reduce” task, where outputs of the “map” task are combined to a reduced key-value pairs, serves as the cornerstone of many big data algorithms and their variations. The paradigm typically include computers used in parallel computations (e.g., hadoop clusters) to be physically clustered in the same location.

Traditional data sciences used in small- or moderate-sized analysis typically require tight coupling of the computations of the “map” and “reduce” steps in a typical big data algorithm. Such an algorithm often executes in a single machine or job and reads all the data at once. How can these algorithms be modified so they can be executed in parallel? If the data is processed in parallel and parsed into subsets, how to leverage the art and science of fusing the results as phrased in the “reduce” step? Making a generic case for a ML/AI algorithm running in a parallel environment proves to be a difficult task. Furthermore, running such an algorithm in a distributed environment is even more challenging, for example, using an agent to compute part of the analysis separately in sub-systems of a system, and then combining the results.

In this paper, we describe a novel infrastructure called collaborative learning agents (CLA) and the application in an operational environment, namely swarm intelligence, where a swarm agent is implemented using a CLA. This infrastructure enables a collection of swarms working together, not only for fusing heterogeneous big data sources in a parallel and distributed fashion, but also for effectively performing customized analytics such as ML/AI algorithms as if they

are as a single agent. We show a use case to use CLA for monitoring the health of a system of systems.

2 Collaborative Learning Agents (CLA)

Our previous work [14] shows the architecture of CLA. A single agent representing a single system capable of ingesting and analyzing data sources while employing a process (i.e., an unsupervised learning process) that separates patterns and anomalies within the data. Multiple agents can work collaboratively in a network. This collaboration is achieved through a peer list defined within each agent, through which each agent passes shared information to its peers. Each agent initially analyzes its own input or content data separately and then fuses the results with its peers'.

In detail shown in Fig. 1, an agent CLA j includes an analytic engine with an algorithm for data fusion and one for ML/AI that can be customized externally. The fusion algorithm integrates the local knowledge base $b(t, j)$ with an input knowledge base $B(t-1, i)$ from its peers i and forms a new knowledge base $B(t, j)$. $B(t-1, i)$ represents all knowledge from i 's network up to point $t-1$. The ML/AI algorithm can be an anomaly detection algorithm, for example, such an algorithm like lexical link analysis (LLA) that assesses the total value of the agent j by separating the new knowledge base $B(t, j)$ into the categories of patterns, emerging and anomalous themes and computes a total value $V(t, j)$ [7, 8]. LLA functions as both a fusion and ML/AI (unsupervised learning) algorithms (see Section 4). A knowledge base $B(t, j)$ contains two components: The first component is an association list which contains pairwise correlations or associations between two word features for structured data or bi-gram word pairs for unstructured data. The second component is a context/concept list, which essentially the same set of context points such as timestamps, geo-locations or file names used in the fusion step to fuse with data from multiple agents.

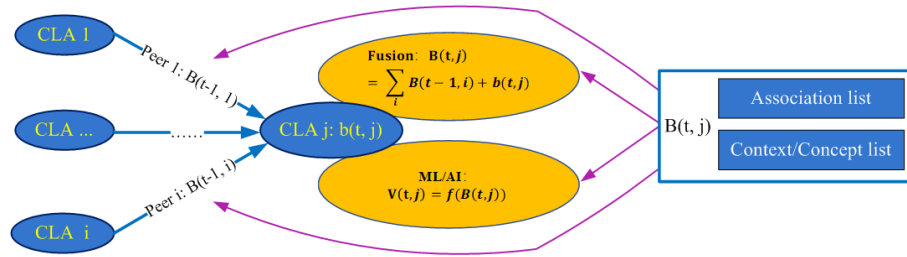


Fig. 1. CLA detail: each agent contains a fusion and ML/AI engine. The fusion is represented as an additive term here as a special case of Step 1 in Section 5. The function forms of the fusion and ML/AI can be customized.

3 Swarm Intelligence (SI)

The CLA concept has an analogue in nature. As human being often ponder: What is the mechanism behind that flocking swarms successfully achieve collective goals such as looking for food or going to places in an optimized fashion using only local and simple communications as shown in Fig. 2 (left) [18]. Often swarms can maximize a total value, e.g., get to a food target in a shortest distance. Swarms find an optimal solution using the pheromone, or the chemical substances produced and released into the environment by a mammal or an insect, which affects the behavior or physiology of others. The concept is simulated in work in AI, i.e., swarm intelligence (SI). SI is the collective behavior of natural or artificial, decentralized and self-organized systems. The expression was introduced in the context of cellular robotic systems as shown in Fig. 2(right) [19].

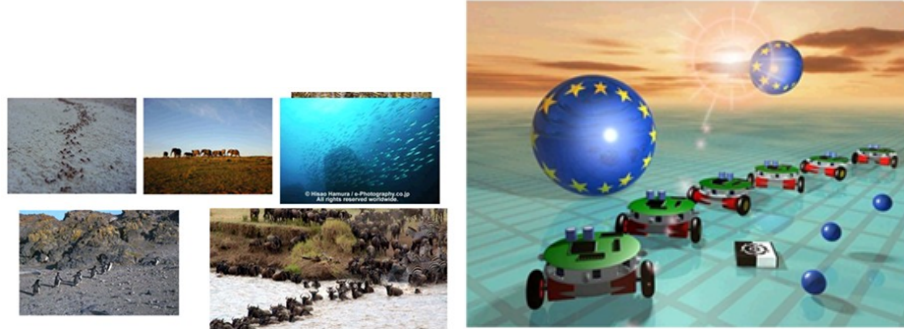


Fig. 2. Left: Natural flocking swarm behaviors [18]. Right: Swarm intelligence has been simulated in the context of cellular robotic systems. It has been used for design armed forces, wireless communications, cellular automata, peer-to-peer networks where the whole system has stronger collective intelligence than individual systems [19]

4 Lexical Link Analysis (LLA) and CLA

4.1 LLA as a Text Analysis Tool for Unstructured Data

In a LLA, a complex system can be expressed in a list of attributes or features with specific vocabularies or lexicon terms to describe its characteristics. LLA is a data-driven text analysis. For example, word pairs or bi-grams as lexical terms can be extracted and learned from a document repository. LLA automatically discovers word pairs, clusters of word pairs and displays them as word pair networks. LLA is related to but significantly different from so called bag-of-words (BOW) methods such as Latent Semantic Analysis (LSA [9], Probabilistic Latent Semantic Analysis (PLSA) [10], WordNet [11], Automap [16], and Latent

Dirichlet Allocation (LDA) [12]. LDA uses a bag of single words (e.g., associations are computed at the word level) to extract concepts and topics. LLA uses bi-gram word pairs as the basis to form word networks and therefore network theory and methods can be readily applied here.

4.2 Extending LLA to Structured Data

Bi-gram also allows LLA to be extended to numerical or categorical data. For example, for structured data such as attributes from databases, we discretize and then categorize attributes and their values to word-like features. The word pair model can further be extended to a context-concept-cluster model [14]. In this model, a context is a word or word feature shared by multiple data sources. A concept is a specific word feature. A context can represent a location, a timestamp or an object (e.g. file name) shared across data sources. In the use case in Section 7, a timestamp is the context.

4.3 Three Categories of High-value Information and Value Metrics

The word pairs in LLA are divided into groups or themes. Each theme is assigned to one of the three categories based on the number of connected word pairs (edges) within a cluster (intra-cluster) and the number of edges between the themes (inter-cluster):

- Authoritative or popular (P) themes: These themes resemble the current search engines ranking measures where information containing the dominant eigenvectors rank high because the elements of the dominant eigenvectors tend to not only connect to each other but also connect to the elements outside a group. They represent the main topics in a data set and are insightful information in three folds: 1) These word pairs are more likely to be shared or cross-validated across multiple diversified domains, so they are considered authoritative; 2) These themes could be less interesting because they are already in the public consensus and awareness, so they are considered popular; 3) The records associated with these themes are considered normal. A popular theme has the largest number of inter-connected word pairs. The content associated with popular themes disseminate faster.
- Emerging (E) themes: These themes tend to become popular or authoritative over time. An emerging theme has the intermediate number of inter-connected word pairs.
- Anomalous (A) themes: These themes may not seem to belong to the data domain as compared to others. They are interesting and could be high-value for further investigation.

Community detection algorithms have been illustrated in Newman [15][16], a quality function (or Q-value), as specifically defined as the “modularity” measure, i.e., the fraction of edges that fall within communities, minus the expected value of the same quantity if edges fall at random without regard for the community structure, is optimized using a “dendrogram” like greedy algorithm. The

Q-value for modularity is normalized between 0 and 1 with 1 to be the best and can be compared across data sets. Formation of the modularity matrix is closely analogous to the covariance matrix whose eigenvectors are the basis for Principal Component Analysis (PCA) [16]. Modularity optimization can be regarded as a PCA for networks. Related methods also include Laplacian matrix of the graph or the admittance matrix and spectral clustering [13]. Newmans modularity assumes a subgraph deviates substantially from its expected total number of edges to be considered anomalous and interesting, therefore, all the clusters or communities (i.e., popular, emerging and anomalous themes regardless) found by the community detection algorithms are considered to be interesting. However, this anomalousness metric does not consider the difference among the communities or clusters.

In LLA, we improve the modularity metric by considering a game-theoretic framework: In a nutshell and in a social network, the most connected nodes are typically considered the most important nodes. However, in LLA, we consider emerging and anomalous information are more interesting and correlated to high-value information. Also, for a piece of information, the combination of popular, emerging and anomalous contributes to the total *value* of the information. Therefore, we define a value metric as follows:

Let the popular, emerging and anomalous value of the information i be $P(i)$, $E(i)$ and $A(i)$ computed from LLA respectively, the total value $V(i)$ for i is defined as in (1).

$$V(i) = P(i) + E(i) + A(i) \quad (1)$$

In the use case in Section 7, we show that the value metrics are correlated with high-value information, e.g., anomalous profiles of Navy engine data.

5 Recursive Learning in CLA and SI

The key advantage of using CLAs relies on using a collection of agents or artificial swarms to perform a task difficult to perform by individual agents. Assume each swarm consists of a CLA and processes part of the total sensor data.

- An agent j represents one sensor or part of the total sensors, operates on its own like a decentralized data analyzer. A single agent does not communicate with all other sensors but only with the ones that are its peers. A peer list is specified by the agent, for example, in Fig. 7, there are three agents in total, CLA 1, CLA 2 and CLA 3. CLA 1 has two peers CLA 2 and CLA 3; CLA 2 has one peer CLA 1; and CLA 3 has one peer CLA 1.
- An agent j collects, analyzes from its domain specific data knowledge base $b(t, j)$. For example, $b(t, j)$ may represent the statistically significant features and associations based on the data observed only by agent j .
- An agent j also includes an analytic engine with two algorithms (i.e., a fusion and ML/AI algorithm) that can be customized externally. We use the two algorithm LLA1 (fusion) and LLA2 (ML/AI) in the implementation of LLA to illustrate the process. The fusion algorithm (LLA1) integrates the

local knowledge base $b(t, j)$ and the global knowledge base $B(t - 1, j)$ into a new knowledge base $B(t, j)$. The ML/AI algorithm (LLA2) assesses the total value of the agent j by separating the total knowledge base into the categories of patterns, emerging and anomalous themes based on the total knowledge base $B(t, j)$ and generates a total value $V(t, j)$. The whole process is displayed as follows:

- Step 1: $B(t, j) = LLA1(B(t - 1, p(j)), b(t, j))$;
- Step 2: $V(t, j) = LLA2(B(t, j))$

Where $p(j)$ represents the peer list of agent j .

- The total value $V(t, j)$ is used in the global sorting and ranking of relevant information.

In this recursive data fusion, the knowledge bases and total values are completely data-driven and automatically discovered and unsupervised-learned from the data. Each agent consists the exact same code, yet collects and analyzes its own data apart from other agents. This agent design has the advantages of decentralized and distributed models: performing learning and fusing simultaneously and in parallel.

6 Fusion and Context Learning in CLA and SI

In a CLA's fusion step, if agent j 's local model $b(t, j)$ shares vocabulary or word features with the knowledge bases that its peers pass to, i.e., $B(t - 1, i)$, the fusion part is simply to modify and update the association list to reflect the new data. Meanwhile, a so-called context learning is performed at each agent using the context/concept list if the agents do not share common word features or vocabulary as follows:

- Step 1: Each agent loops through Peer i in its peer list, and list all contexts and associations from its peers and local data $b(t, j)$.
- Step 2: For each concept (word feature) i_c in $B(t - 1, i)$, check if agent j 's local data $b(t, j)$ and concept j_c to see if it has the same context. If yes, concept i_c and concept j_c in agent i and agent j is linked and the association is added to the knowledge base $B(t, j)$

Fig. 1 also shows the update algorithm and context learning algorithm. Both are part of the whole fusion algorithm (LLA1). The LLA2 refers the part of the LLA algorithm that categorize word features into popular, emerging and anomalous ones.

7 Use Case

At the heart of the US Navy are thousands of machines that drive the ships and submarines. The U.S. Navy mission is to maintain, train and equip combat ready naval forces capable of winning wars, deterring aggression and maintaining freedom of the seas. The US Navy needs to harness big data, data sciences, and

ML/AI to better understand these machines as system of systems. A test data set was culled from the Navy’s engine rooms around the world, link data sciences, ML/AI. The data was used in the Hack The Machine event in Cambridge, Massachusetts in September, 2017 organized by the US Navy [17].

The data set is a typical health maintenance use case where multiple sensors are used to monitor a system of systems (e.g., a ship) to see if it is operated normal or if there are any “health” issues. The sensor data collected can be in a variety of heterogeneous formats such as numerical values, image and text etc. The correlations and associations of the multiple sensor data are not necessarily known before the data collection. The sensors can also be installed in a distributed fashion, for example, in different ships or in the same ship but different ship subsystems.

Fig. 3 is a ML/AI paradigm to learn from historical data of system of systems (e.g., ship 1 and ship 2) and then apply the knowledge patterns learned and discovered to the new data (e.g., ship-x). The CLAs in a swarm intelligence can reside in the systems or subsystems in a distributed fashion in this case.

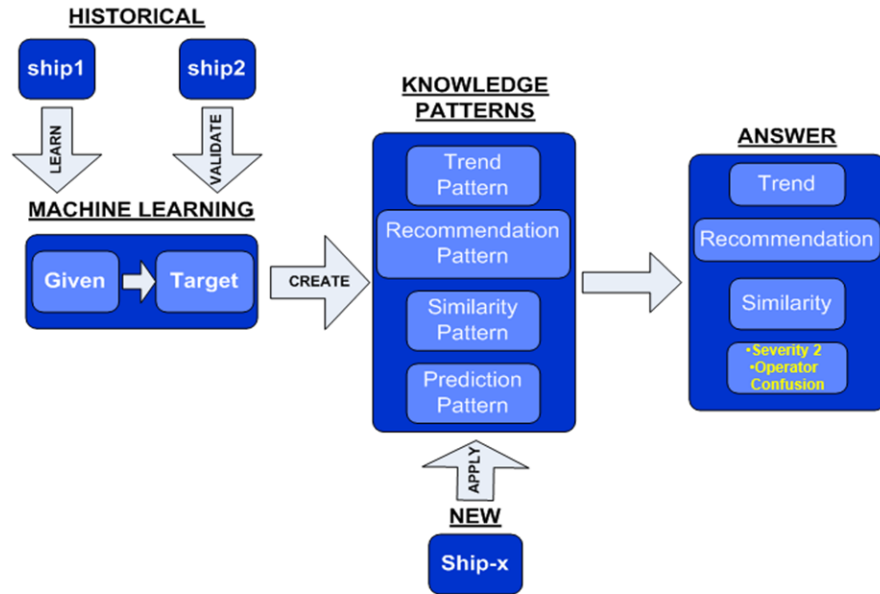


Fig. 3. Data sciences, ML/AI meet to check the health of a system of systems

Fig. 4 shows an sample of the original data set with about 50 variables with timestamps over a period of a year (7/2016 to 6/2017). Each numerical variable is discretized into LLA word features based on the initial statistics such as means and standard deviations. The total LLA features $n = 160$. For example, *bearing_temp_aft_bt_107.97_136.70* represents a feature from the

original sensor measurement variable *bearing_temp_aft* with its value between 107.97 and 136.70, where 107.97 and 136.70 is the mean and the mean plus one standard deviation. The values were generated automatically and initially within a CLA. If only one CLA is used with the fusion of a set of agents (swarms), in order to perform the ML/AI in the second step, the association list $B(t, j)$ needs to be computed with the amount of the computation $O(n^2m)$, where m is the number of contexts (i.e., timestamps) that can link these word features.

[illegible]

Fig. 4. An example from the original sensor data set with about 50 variables with timestamps over a period of a year (7/2016 to 6/2017)

Three results were discovered by a single CLA as follows:

- Three clusters were discovered by a single CLA as shown in Fig. 5.
 - Two green clusters represent normal running conditions.
 - Blue clusters represents outliers findings (anomalies).
- Characteristics of the anomaly cluster 1: Class A, Ship #1 with all gas turbine generator data
- If Turbine Inlet Temp. $> 409.60^{\circ}F$, then the blue parts/units have a high likelihood to fail (anomalies) in near future and should be checked.

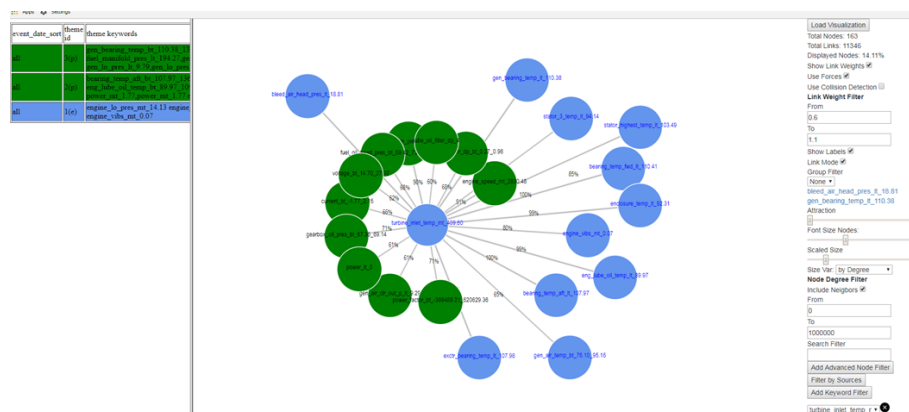


Fig. 5. Two green clusters represent normal running conditions

For this maintenance sensor data, the CLA generated 160 word features from the 50 sensors. The identified features as shown highlighted in blue in Fig. 8 out of the total 160 ones are more sensitive to the engine operating performance.

Fig. 6 shows all the time series for selected variables in the anomaly group distributed along time points when all the sensor data are processed together, i.e., in one CLA. The anomaly time points are shown in blue dots which have higher emerging scores in the y-axis.

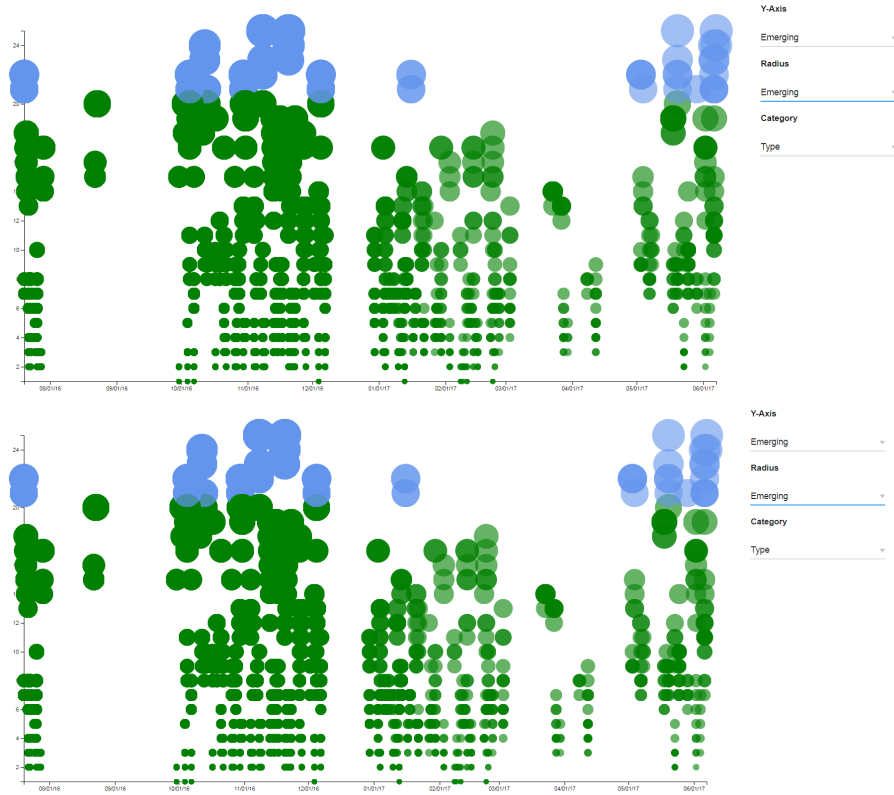


Fig. 6. Top: All three groups shown in a time series relationship with the anomaly time points when all the sensor data are together, i.e., in one CLA. Bottom: All three groups shown in a time series relationship with the anomaly time points in three separate CLAs and then fused together

To illustrate the use of SI, we divided the 160 features into three groups and each set of the features and associated data are processed separately in three CLAs as shown in Fig. 7 (left). Fig. 7 (right) shows the peer lists for three agents for themselves. The agents do not have to fully connected to each other. Each agent periodically performs the algorithms LLA1 and LLA2, and the whole

system converges to an equilibrium state where every agent acquires the same global knowledge base. Each agent can also decide not to publish some of its knowledge base $B(t, j)$. In this case, we may call the agent possesses private information or retains expertise for itself.

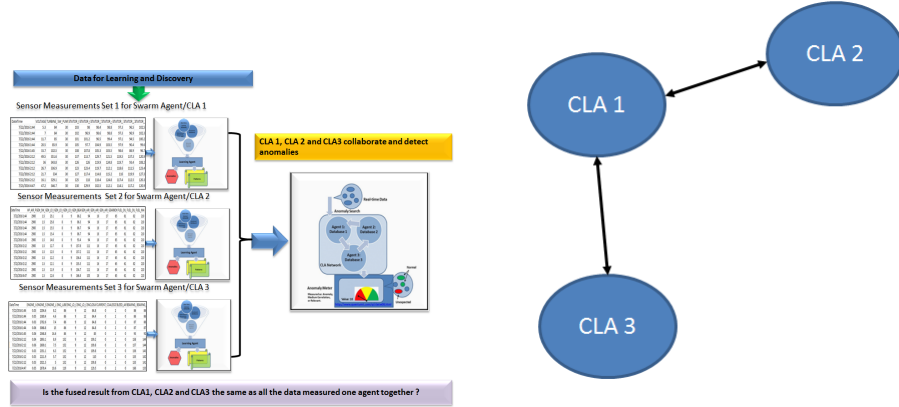


Fig. 7. Left: SI is shown in three CLAs. Right: Three CLAs do not have to be fully connected. The knowledge bases “spread out” over many iterations of the fusion and ML/AI algorithms LLA1 and LLA2

As shown in Fig. 6, the swarm CLAs (bottom) generate the identical results and time series visualization as if all the sensors in one swarm (top). The swarm agents can compute the exact same fusion and ML/AI computation as if the data is collected and processed in a single system. Therefore, the total correlation computations $O(n^2m)$ is distributed and decentralized among three agents.

8 Conclusion

In this paper, we showed how collaborative learning agents and swarm intelligence are used to analyze data from a system of systems. We showed an application and a use case of quickly examining the health and maintenance issues of a sample Navy ship which might be used to generate early warning and recommendations.

A single agent/swarm is able to identify features that are anomalous and more sensitive to the engine performance. Multiple agents/swarms collaborate to distribute and decentralize the computation as if the data and computation are collected and analyzed all together in a single system.

The mechanism described in this paper is not a simple map/reduce mechanism or a collection of parallel processes because the collective behavior of swarm agents are iterated and converge towards to a stable state as if all the big data are

CLAS word features	Percentage of records	CLAS word features	Percentage of records	CLAS word features	Percentage of records
lube_oil_filter_dp_0_5	0%	0% gen_sw_stnrv_dp_bt_0_27	0%	0% current_mt_2_15	0%
lube_oil_filter_dp_7	1%	1% hp_air_pres_mt_2975.81	1%	0% bearing_temp_fwd_mt_168.11	1%
turbine_inlet_temp_bt_171.42_290.51	1%	1% gen_bearing_temp_mt_168.14	1%	1% bleed_air_head_pres_bt_18.81_52.87	1%
turbine_inlet_temp_mt_409.60	2%	2% excr_bearing_temp_mt_165.42	2%	2% bearing_temp_aft_mt_165.49	2%
lube_oil_filter_dp_5	2%	2% excr_bearing_temp_bt_107.97_136.70	2%	2% bearing_temp_aft_bt_107.97_136.70	2%
power_mt_1.77	2%	2% gen_bearing_temp_bt_110.38_139.26	2%	3% bearing_temp_fwd_bt_110.41_139.26	3%
power_factor_mt_620629.36	2%	2% gen_air_temp_mt_114.22	2%	4% bleed_air_head_pres_mt_86.93	4%
lube_oil_filter_dp_5	6%	6% fuel_oil_head_pres_mt_73.81	6%	6% eng_lube_oil_temp_bt_89.97_109.97	9%
sw_pump_dis_pres_mt_43.76	13%	13% fuel_filter_dp_bt_0.54	13%	7% engine_lo_pres_bt_6.47	11%
stator_3_temp_mt_117.90	16%	16% gen_lo_clr_out_pres_bt_3.21_7.64	16%	9% eng_lo_clr_out_pres_mt_10.88	11%
stator_6_temp_mt_120.74	17%	17% gen_air_clr_out_p_bt_9.29_16.58	17%	9% eng_lube_oil_temp_mt_129.97	12%
stator_1_temp_mt_124.48	17%	17% fuel_oil_head_pres_bt_45.03	17%	9% coalescer_diff_pres_bt_0.57	13%
stator_highest_temp_mt_130.35	17%	17% gen_lo_pres_bt_9.79	9%	9% engine_lo_pres_mt_14.13	13%
stator_2_temp_mt_124.13	17%	17% gen_lo_clr_in_pres_mt_15.23	10%	10% engine_vibs_mt_0.07	16%
voltage_bt_14.70	17%	17% gearbox_oil_pres_bt_65.58_67.36	11%	11% eng_lo_clr_in_pres_bt_5.22_11.26	16%
stator_4_temp_mt_117.06	19%	19% fuel_oil_head_temp_mt_97.46	12%	12% coalescer_diff_pres_mt_0.05	17%
stator_4_temp_bt_84.64	19%	19% gen_air_clr_in_p_mt_26.50	12%	12% current_bt_0	21%
power_factor_bt_0	19%	19% gen_air_clr_out_p_mt_23.87	12%	12% eng_lo_clr_out_pres_bt_3.22	21%
stator_5_temp_mt_120.61	19%	19% gen_lo_clr_out_pres_mt_12.07	12%	12% bleed_air_head_pres_bt_52.87_86.93	22%
stator_5_temp_bt_94.19	21%	21% fuel_manifold_pres_bt_194.27	17%	17% eng_lo_clr_out_pres_bt_2.21_7.05	22%
stator_3_temp_bt_94.14	21%	21% gearbox_oil_pres_mt_69.14	17%	17% enclosure_temp_mt_153.41	23%
stator_6_temp_bt_92.22	21%	21% gen_air_clr_in_p_bt_13.38	17%	17% eng_lo_clr_in_pres_mt_11.26	26%
stator_1_temp_bt_96.92	21%	21% fuel_filter_dp_mt_3.42	18%	18% enclosure_temp_bt_92.31	27%
stator_2_temp_bt_94.97	21%	21% frequency_bt_0	18%	18% eng_lube_oil_temp_bt_89.97	27%
lube_oil_filter_dp_1	23%	23% fuel_oil_head_temp_bt_86.94	18%	18% bearing_temp_fwd_bt_110.41	27%
stator_highest_temp_bt_103.49	23%	23% gen_lo_pres_bt_13.45_17.11	19%	19% bearing_temp_aft_bt_107.97	28%
lube_oil_filter_dp_3	24%	24% fuel_manifold_pres_mt_236.41	19%	19% engine_speed_bt_2218.24	28%
voltage_mt_41.14	24%	24% gen_lo_clr_in_pres_bt_5.19_10.21	20%	20% engine_speed_mt_2830.48	28%
turbine_inlet_temp_bt_171.42	25%	25% gen_sw_stnrv_dp_mt_1.65	21%	21% enclosure_temp_bt_92.31_122.86	28%
sw_pump_dis_pres_bt_31.26	26%	26% hp_air_pres_bt_2263.07	23%	23% engine_vibs_bt_0.03_0.05	30%
sw_pump_dis_pres_bt_37.51_43.76	27%	27% gen_air_temp_bt_76.10	23%	23% engine_vibs_bt_0.03	31%
lube_oil_filter_dp_4	29%	29% hp_air_pres_bt_2263.07_2619.44	27%	27% engine_speed_bt_2524.36_2830.48	33%
voltage_bt_27.92_41.14	30%	30% gen_lo_pres_mt_17.11	28%	28% engine_speed_bt_2218.24_2524.36	33%
stator_highest_temp_mt_103.49_116.92	32%	32% frequency_bt_0.02_0.12	29%	29% engine_lo_pres_bt_10.30_14.13	33%
stator_3_temp_bt_94.14_106.02	32%	32% gen_bearing_temp_bt_110.38	30%	30% coalescer_diff_pres_bt_4.81_9.05	36%
stator_1_temp_bt_96.82_110.70	35%	35% excr_bearing_temp_bt_107.98	33%	33% enclosure_temp_bt_122.86_153.41	43%
stator_2_temp_bt_94.97_109.55	35%	35% gen_lo_clr_in_pres_bt_5.19	33%	33% engine_vibs_bt_0.05_0.07	45%
lube_oil_filter_dp_2	36%	36% gen_lo_clr_out_pres_bt_3.21	34%	34% coalescer_diff_pres_bt_0.57_4.81	54%
stator_6_temp_bt_92.22_106.48	37%	37% gearbox_oil_pres_bt_65.58	34%	34% engine_lo_pres_bt_6.47_10.30	64%
stator_5_temp_bt_94.19_107.40	39%	39% gen_air_temp_bt_76.10_95.16	34%	34% eng_lo_clr_out_pres_bt_7.05_10.88	67%
stator_4_temp_bt_84.64_100.85	40%	40% gen_sw_stnrv_dp_bt_0.96_1.65	36%	36% eng_lube_oil_temp_bt_109.97_129.97	74%
stator_5_temp_bt_107.40_120.61	43%	43% gen_air_clr_out_p_bt_9.29	39%	39% eng_lo_clr_in_pres_bt_0.5_2.2	80%
stator_4_temp_bt_100.85_117.06	44%	44% fuel_manifold_pres_bt_194.27_225.34	39%	39% bearing_temp_aft_bt_107.97_165.43	90%
stator_6_temp_bt_106.48_120.74	46%	46% fuel_oil_head_temp_bt_86.94_92.20	45%	45% bearing_temp_fwd_bt_139.26_168.11	91%
stator_2_temp_bt_109.55_124.13	48%	48% fuel_filter_dp_bt_1.98_3.42	52%	52% bleed_air_head_pres_bt_18.81	95%
stator_1_temp_bt_110.70_124.48	48%	48% gen_air_clr_in_p_bt_29.94_26.50	52%	52% current_bt_1.77_2.15	100%
stator_highest_temp_bt_116.92_130.35	49%	49% gen_air_clr_in_p_bt_23.38_19.94	54%		
voltage_bt_14.70_27.92	50%	50% fuel_oil_head_pres_bt_45.03_59.42	50%		
stator_3_temp_bt_106.02_117.90	52%	52% fuel_filter_dp_bt_0.54_1.98	59%		
sw_pump_dis_pres_bt_31.26_37.51	55%	55% fuel_oil_head_temp_bt_92.20_97.46	60%		
power_bt_0	56%	56% fuel_manifold_pres_bt_225.34_255.41	60%		
power_bt_6.82_1.77	62%	62% fuel_oil_head_pres_bt_59.42_73.81	61%		
turbine_inlet_temp_bt_290.51_409.60	94%	94% gen_lo_clr_in_pres_bt_10.21_15.23	72%		
power_factor_bt_399489.21_620629.36	100%	100% gearbox_oil_pres_bt_67.36_69.14	73%		
		gen_air_temp_bt_95.16_114.22	74%		
		gen_air_clr_out_p_bt_16.58_23.87	75%		
		gen_sw_stnrv_dp_bt_0.27_0.96	78%		
		gen_lo_pres_bt_9.79_13.45	79%		
		gen_lo_clr_out_pres_bt_7.64_12.07	80%		

Fig. 8. Variables in all three swarms and highlighted variables are more important and sensitive in the emerging groups

processed in a single swarm. Each swarm actually follows a game-theoretic dual process of finding an equilibrium for itself and meanwhile achieving the maximum social welfare for the whole system. The final state of swarms is decided not only by the data collected individually but also by how the data of different agents interact, correlate, and associate with each other, just like a community of swarms, social animals and humans. An agent has one or multiple types of expertise as special data for itself. Agents share knowledge and collaborate when applying different types of expertise.

Each agent possesses the exact same code but analyzes different (sensor) data from each other. This agent design has the advantages for decentralized and distributed computing, performing learning and fusion simultaneously and in parallel as in the internet of things (IoT). Swarm intelligence is an important aspect of IoT systems or broadly defined system of systems.

9 Acknowledgements

Thanks to the “Hack The Machine” event (<https://www.hackthemachine.ai>) at the Massachusetts Institute of Technology (MIT), Cambridge, MA, organized and sponsored by the US Navy. The views and conclusions contained in this presentation are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the U.S. Government.

References

1. Girvan, M., and Newman, M. E. J.: Community structure in social and biological networks. In: Proceedings of the National Academy of Sciences, USA, 99(12), 7821-7826, 2002.
2. Freeman, L.C.: Centrality in social networks I: conceptual clarification. *Social Networks*, 1: 215-239 (1979)
3. Brin, S. and Page, L.: The Anatomy of a large-scale hypertextual web search Engine. *Computer Networks and ISDN Systems*, 30:107-117 (1998)
4. Kleinberg, J. M.: Authoritative sources in a hyperlinked environment, *Journal of the ACM* 46 (5): 604, 1999. Retrieved from http://en.wikipedia.org/wiki/HITS_algorithm
5. Mihakcea, R. and Tarau, P.: TextRank: bringing order into texts. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), July 2004, Barcelona, Spain
6. Erkan, G. and Radev, D. R.: Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22:457-479 (2004)
7. Zhao, Y., Gallup, S.P. and MacKinnon, D.J.: System self-awareness and related methods for improving the use and understanding of data within DoD. *Software Quality Professional*, 13(4): 19-31 (2011). Retrieve from <http://asq.org/pub/sqp/>
8. Zhao, Y., Mackinnon, D. J., Gallup, S. P.: Big data and deep learning for understanding DoD data, *Journal of Defense Software Engineering. Special Issue: Data Mining and Metrics* (2015)

9. Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S.: Using latent semantic analysis to improve information retrieval. In: Proceedings of CHI88: Conference on Human Factors in Computing, 281-285 (1988)
10. Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden (1999)
11. Miller, G. A.: WordNet: a lexical database for English., Communications of the ACM, 38(11)(1995)
12. Blei, D., Ng, A. and Jordan, M.: Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993-1022 (2003). Retrieved from <http://jmlr.csail.mit.edu/papers/volume3/blei03a/blei03a.pdf>
13. Ng, A., Jordan, M., and Weiss, Y.: On spectral clustering: analysis and an algorithm. In: T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14 (pp. 849-856), (2002). MIT Press. Retrieved from <http://ai.stanford.edu/~ang/papers/nips01-spectral.pdf>
14. US patent 8,903,756: System and method for knowledge pattern search from networked agents (2014). Retrieved from <https://www.google.com/patents/US8903756>
15. Newman, M. E. J.: Fast algorithm for detecting community structure in networks (2003). Retrieved from <http://arxiv.org/pdf/cond-mat/0309508.pdf>
16. Newman, M. E. J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E, vol. 74, no. 3 (2006)
17. HackTheMachine (2017) <https://www.hackthemachine.ai/level-project/>
18. Perretto, M. and Lopes, H. S.: Reconstruction of phylogenetic trees using the ant colony optimization paradigm. Genet. Mol. Res. 4 (3): 581-589 (2005). Retrieved from http://www.funpecrp.com.br/gmr/year2005/vol3-4/wob09_full_text.htm
19. Beni, G. and Wang, J.: Swarm intelligence in cellular robotic systems. Robots and Biological Systems: Towards a New Bionics? Springer Berlin Heidelberg, 703-712 (1993).