Self-learning Data Processing Framework Based on Computational Intelligence

Enhancing Autonomous Control by Machine Intelligence

Prapa Rattadilok
School of Computing Science and Digital Media
Robert Gordon University
Aberdeen, UK
p.rattadilok@rgu.ac.uk

Andrei Petrovski
School of Computing Science and Digital Media
Robert Gordon University
Aberdeen, UK
a.petrovski@rgu.ac.uk

***Abstract*—A generic framework for evolving and autonomously controlled systems has been developed and evaluated in this paper. A three-phase approach aimed at identification, classification of anomalous data and at prediction of its consequences is applied to processing sensory inputs from multiple data sources. An ad-hoc activation of sensors and processing of data minimises the quantity of data that needs to be analysed at any one time. Adaptability and autonomy are achieved through the combined use of statistical analysis, computational intelligence and clustering techniques. A genetic algorithm is used to optimise the choice of data sources, the type and characteristics of the analysis undertaken. The experimental results have demonstrated that the framework is generally applicable to various problem domains and reasonable performance is achieved in terms of computational intelligence accuracy rate. Online learning can also be used to dynamically adapt the system in near real time.**

Keywords—computational intelligence; evolving and autonomous systems; anomalies; robot controls

I. Introduction

There is a growing demand for intelligent and autonomous control in engineering applications in order to achieve a more reliable and appropriate operations in an uncertain environments [1]. This is especially true when some constraints are present that cannot be satisfied by human intervention with regard to decision-making speed in life threatening situations (e.g. automatic collision systems, exploring hazardous environments) or to process large volumes of data). Because machines are capable of processing large amounts of heterogeneous data much faster and are not subject to the same level of fatigue as humans, the use of computer-assisted control in many practical situations is preferable.

Data obtained from sensors is commonly used for robot control and navigation. Due to the fast pace of computational advancements, the quality of the data obtained from the sensors as well as the quantity of the sensors themselves have increased dramatically. The vast availability of data that can be measured has shifted recent research focus onto another challenging task of intelligently processing this data and inferring useful information. These inferences may be used to dynamically control relevant computing processes (e.g. evaluating the reliability of the data obtained) which may be used to control different computing processes (e.g. activate another sensor, collect additional data to verify the problem) or operate connected physical entities (e.g. manoeuvre the robot away from obstructions). The complexity of this task increases exponentially especially in a real-time automated process control scenario. The information about a possible failure is generally more useful before the failure takes place, especially when prevention and damage control can be carried out in order to either completely avoid the failure, or at least alleviate its consequences.

Computational Intelligence (CI) techniques have been successfully applied in various application domains [2]. These techniques however require training data to provide reliable and reasonably accurate specification of the context in which the system operates. The context or pattern(s) of interest enables the system to highlight potential anomalies in the data so that intelligent and autonomous control of the underlying process can be carried out.

Anomalies are defined as incidences or occurrences, under a given circumstances or a set of assumptions, that are different from the expectance. By their nature, these incidences are rare and often not known in advance. This makes it difficult for the computational intelligence techniques to form an appropriate training dataset. Moreover, dynamic problem environments can further aggravate the lack of training data by the increased likelihood of intermittent anomalies. Computational intelligence techniques that are used to tackle dynamic problems should therefore be able to adapt to environmental/contextual changes.

In this paper, a multi-tiered framework for evolving, autonomous learning systems with heterogeneous input sources has been developed and evaluated on robot navigation data. The framework is aimed at tackling problems in a real-time input data stream in dynamic problem environments, where the pattern(s) of interest (e.g. anomalies or faults) are relatively new and data related to them are limited or unknown. In order to achieve this, both statistical analysis and computational intelligence techniques are applied within the framework together with the online learning capability that allows for adaptive and autonomic problem solving. The framework has been successfully applied in the field of traffic surveillance [3], smart home environment as well as automotive process control [4]. This paper reports the evaluations on the applicability of the framework, on a different problem domain with minimal adjustments. The framework is used to evaluate the data obtained from the sensors: identify anomalies in this data and confirm the trustworthiness of these anomalies by evaluating data obtained from additional sensors.

II. Data-Driven Framework for Evolving and Autonomous Leaning Systems

The integration between computing processes and physical entities creates a dynamic problem environment of a network of interacting elements. This concept is prevalent in the field of sensor networks and robotics. Data about the real world environment can be transmitted to the computing processes within the system through monitoring and sensing; while inferences, which reflect policies of the system, are obtained by enabling the intelligent processing of these input data streams, and are used to autonomously control other computing processes or operate connected physical entities in the real world [5]. These interactions between computing processes and physical entities may results in changes in the robot's real world environment and therefore dynamically alter the resulting computing processes and operations of physical entities.

Fig. 1. illustrates the multi-tiered data driven framework for evolving and autonomous learning systems [3], where each tier or layer is dedicated to a certain context processing task, ranging from low-level context acquisition up to high-level context application using either existing or acquired knowledge.

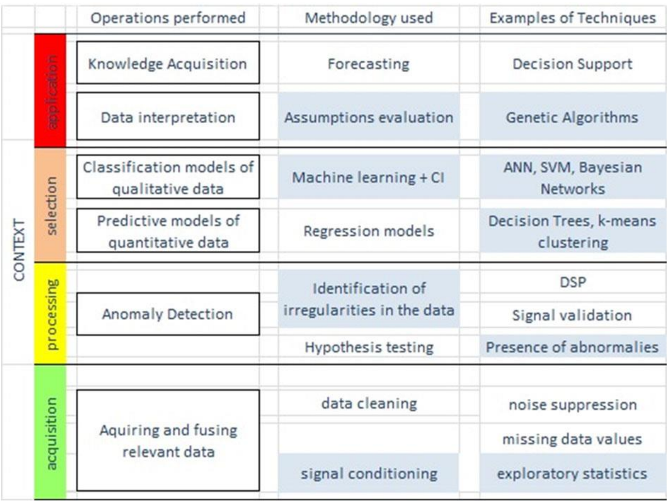

Fig. 1. A data-driven framework for building autonomous learning systems

The framework in Fig. 1 spans a number of well-known and well-established disciplines that include machine learning, system identification, data mining, computational intelligence, signal processing, control theory and pattern recognition. The interconnections between computing processes and physical entities that must instantaneously exchange, parse and act upon heterogeneous data in a coordinated way, creates two major challenges: how best to process the amount of data available from various data sources that need to be processed at any given time and the choice of computing processes or physical entity configuration in response to the information obtained from the data collected and analysed. Fig. 2 illustrates a systematic approach to handling the challenges related to the volume of data, its veracity and velocity by optimising the balance between data availability and its quality.

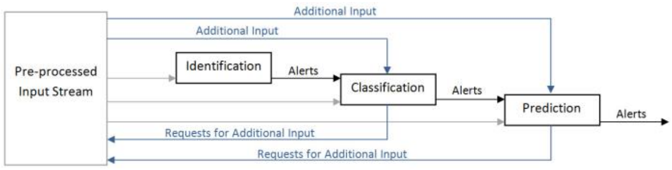

Fig. 2. Systematic approach to handling data and computing processes

The processing of the input data stream is segregated into identification, classification and prediction phases; which corresponds to the two middle layers in Fig.1 (i.e. Processing and Selection layers). The identification phase minimises the volume of data and the data processing cost by analysing only inputs from easy to process data sources using anomaly identification techniques such as outlier detection or other statistical analyses. Identified potential anomalies are then passed onto the classification phase and classified into different types. At the end of the process, the prediction phase examines the consequences of the discovered anomalies being present in the operations of the underlying system.

Such an approach allows for the acquisition of data and/or activation of the necessary physical entities on an ad-hoc basis, depending on the outcome at each phase. Moreover, the accuracy attained at the specified phases can be enhanced by incorporating additional data from alternative sources.

III. Processing Anomalies where the Pattern(s) of Interest is Unknown

Computational intelligence techniques and expert systems have been successfully applied to tackling many anomaly detection problems where patterns of interest are known. Anomalies are rare and often incidences that are unexpected, and/or deviate excessively from the norm. Detecting anomalies with unknown pattern(s) of interest is even more complicated. Statistical analyses and clustering are examples of techniques that are commonly used to tackle such characterisations [6]. Fig. 3 illustrates a more detailed process for the systematic approach depicted in Fig. 2 where statistical analysis and computational intelligence techniques are combined to autonomously identify suitable pattern(s) of interest for anomalies in the input data stream and learn from the experience when similar anomalies occur again. The approach is aimed at dealing with real-time systems that integrate a number of computing processes and physical entities. The logistical challenges of controlling physical entities is tackled in an ad-hoc basis, by executing required computing processes or activating necessary physical entities to obtain relevant data and processed required information when needed.

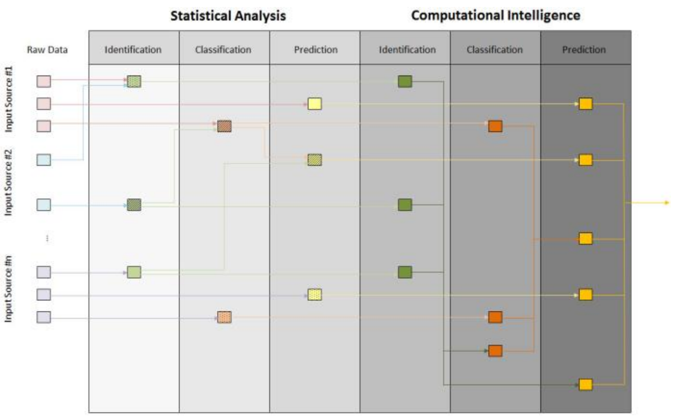

Fig. 3. Systematic approach to process anomalies where data related to the pattern(s) of interest is limited or unknown

From Fig. 3, "Input Sources" can be homogenous or heterogeneous sensors (e.g. ultrasound, thermal); or it can be computing processes (e.g. performing calculations, activating additional sensors); or even physical entities (e.g. changing the robot's direction and manoeuvres away from the obstruction). The "Statistical Analysis" refers to STATISTICAL FUNCTIONS and *time windows*, for example, an AVERAGE (of data from an input source) over *the last 5 seconds*. The time windows can be continuous (e.g. time t_0-t_{10}, time t_1-t_{11} and so on); or in consecutive but non-repetitive chunks (e.g. time t_0-t_{10}, time t_{11}-t_{20} and so on); or in separation distances (e.g. t_0 and t_{10}; t_1 and t_{11}). A statistical analysis can also be nested, for example an AVERAGE of DIFFERENCE between two input values measured at *two different time steps having a time window of 10 in between the two measurements*.

In the Identification column, the dotted light green square (the one on the top) may represent an AVERAGE of the distance between the robot and the wall *over the last 5 seconds*. An ultrasound sensor may be used to measure the distance between the robot and the wall, this is represented by the pink squares Input Source #1 in Fig. 3. The slanted light green square (the one in the middle of the identification column) may represent a DIFFERENCE of the distance between the robot and the wall *from the previous measurement*. A different ultrasound sensor (e.g. on the opposing direction to the first sensor) may be used to measure this distance, this is represented by the blue squares Input Source #2. The non-patterned light green square (the one at the bottom of the identification column) may represent the distance between the robot and the wall without applying any form of statistical analysis. These three light green squares in combined may represent a collection of statistical analyses applied to data gathered from different input sources that can be used to represent a pattern of interest (e.g. whether or not a data point is anomalous).

When the problem space is well defined or when there is a clear definition of the pattern(s) of interest, the decisions about which input sources and data streams to use; which statistical analyses are used to process these input data streams; and which computational intelligence techniques are used to identify the pattern(s) of interest, are in most cases, made by human experts through knowledge and experimentations.

In terms of data processing, the combinations of statistical analyses and time windows can be viewed as adding different filters to input data streams. Processed data is passed to different learning techniques and used to in different phases. This framework extends to include Classification and Prediction phases, where output from previous phases can be used in the later phases. For example, the abrupt change in distance may cause the identification phase to identify the data point as anomalous. The classification phase may then attempt to classify the cause of such changes as internal (e.g. the sensor itself is malfunction) or environmental (e.g. an obstruction) by evaluating and validating the anomalous data point with data from additional sensors. If the classification phase verified the anomaly and confirmed an obstruction, the prediction phase may activate additional sensors, search for an optimal path and manoeuvre the robot away from the obstruction.

Fig. 4 illustrates an evolving and autonomous learning process for the systems when the problem space is not well defined by exploiting a meta-learning algorithm. In this case a Genetic Algorithm is used as the context-application controller that autonomously learns and makes decisions about these choices for techniques operating at the lower layers.

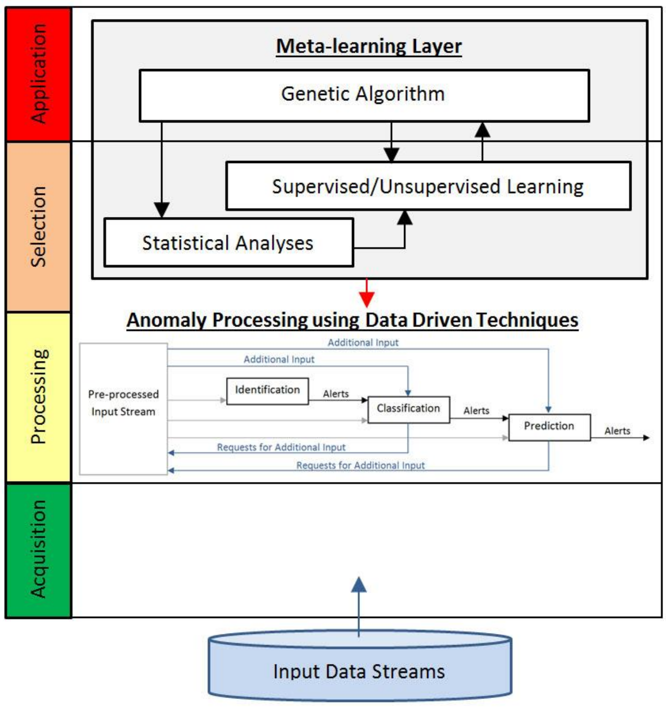

Fig. 4. A data-driven multi-tiered framework for evolving and autonomous learning systems by applying genetic algorithm

The genetic algorithm, shown in Fig. 4, can be used to autonomously evolve system's parameters (e.g. input sources to use, statistical analyses to process the data streams, supervised or unsupervised learning techniques to be used), a task which is generally carried out empirically by practitioners. The parameter optimisation can extend to encompass technique specific parameter settings. Using an artificial neural network as an example, typical examples of technique specific parameters are the number of input, hidden and output nodes in the artificial neural network, acceptable error rates, and stopping criteria.

After the meta-learning process has been carried out, the tuned learning technique(s) can be used to directly analyse the input data stream in order to provide the anomaly identification, classification, and ultimately prediction, capabilities. If required, this tuning process can be carried out continuously and in near-real time.

IV. Experimental Results

The data-driven multi-tiered framework for evolving and autonomous learning systems is implemented in JAVA and the Encog machine learning library [7]. Artificial Neural Network (ANN), Support Vector Machine (SVM), Bayesian Network (Bayes) are the computational intelligence techniques currently available in the framework, together with the K-Means clustering technique. Built in statistical analyses include Difference, Average, Variance, Standard deviation, Summation, Min and Max.

A. Anomalies in the Robot's Sensor Data Streams

The data is collected from a robot navigating through a room following the wall in a clockwise direction for four circuits, using 24 ultrasound sensors arranged circularly around its waist, each directed at 15° angle increments [8]. Sensor readings are sampled at a rate of 9 samples per second. Each value in the dataset indicates the distance between the robot to the wall at different time steps – the bigger the value, the further the robot is from the wall in the direction that that sensor is pointing at.

The data stream passes through the three phases approach described in Fig. 2. The potential anomalies are picked up during the identification phase. Averages of four different time window lengths are used – 5, 10, 20 and 50 time steps to identify any abrupt changes in the data stream. A data point is identified as a potential anomaly when the current value is 3 times higher or 50 times lower than the moving averages of at least two different time window lengths.

By using statistical analyses, any abrupt changes in the input data stream are flagged before passing onto the classification phase. The idea is to quickly flag up a potential anomaly at the identification phase by evaluating only one sensor input rather than processing the data from all 24 sensors. During this second phase, these potential anomalies are validated or the level of signal interference is estimated by using additional data from adjacent sensors. There are two possible main outcomes during at classification phase: either the anomalies are confirmed or invalidated. The tasks of the prediction phase vary depending on the outcome of the classification phase.

If the readings from the adjacent sensors do not corroborate the presence of an anomaly in terms of the measured distance, the result of the classification phase would be invalidation of the original sensor readings. For example, if sensors A, B and C are facing the directions of -15°, 0° and 15° angle respectively from the direction to the wall, and if the change in distance of A and C is equal to 1 while the change in distance of B is equal to 5, it can be assumed that sensor B may be malfunctioning. As a result of this, the prediction component will interpolate a corrected value from the other two sensors, and keeping track of the frequency of the malfunction in order to alert the user about a potential breakdown of the equipment.

On the other hand, if the sensor is performing correctly and the other sensors confirmed the abrupt changes. The most recent operation that was performed (i.e. propagating, turning or stopping the robot) is recorded as this operation must be causing the abrupt change. The prediction phase might then be used to predict the outcome of continuing this operation, and if the outcome is undesirable, the operation will be automatically changed (e.g. switching maneuvering directions).

B. Offline Learning

The developed framework is aimed at anomalies with limited data related to the patterns of interest. The system therefore is initiated with a simple anomaly identification process using statistical analyses (i.e. moving average with different time windows). These potential anomalies are labelled and used for the learning purpose of different computational intelligence techniques. Given the feedbacks from human users, the system gradually and automatically switches from simple statistical analyses to more sophisticated pattern recognitions.

Generally, computational intelligence techniques perform differently on various real life problems. The most obvious distinction between the techniques is the optimal balance between accuracy of prediction and training speed. The rationale behind having a number of computational intelligence techniques available in the framework is to allow the system to automatically apply the best technique to the current problem solving stage. Table 1 summarises the number of false negatives and false positives (shown as their ratio) when various computational intelligence techniques are applied to the dataset. The performance of the developed framework is compared with that of WEKA [9].

In this context, false negatives are defined as the data points identified by the statistical analysis node as being anomalous (or normal in the case of false positives), but the computational intelligence techniques failed to identify them as such.

TABLE I. THE NUMBERS OF FALSE NEGATIVES AND FALSE POSITIVES FOR DIFFERENT COMPUTATIONAL INTELLIGENCE TECHNIQUES (A) ON THE UNBALANCED DATASET AND (B) ON THE BALANCED DATASET

	Multi-tiered		**WEKA**	
	(a)	***(b)***	***(a)***	***(b)***
ANN	3/6	1/0	7/3	0/1
SVM	14/3	0/0	13/7	0/1
BAYES	72/0	0/0	17/3	0/0
K-Means	72/0	17/0	69/1104	19/2

When an unbalanced dataset (i.e. with a significant difference in the size of normal and abnormal data sets) was used (columns (a)), the proposed framework was unable to correctly identify any of the 72 anomalous data points with either a Bayesian network or K-Means clustering. At the same time, when using WEKA, over 1000 data points were incorrectly clustered by K-Means clustering. When a balanced dataset (i.e. when an equal number of anomalous and normal data points are presented in the training dataset) was used (columns (b)), the number of incorrectly identified data points is greatly reduced – in particular, the support vector machine and Bayesian network (BAYES) within the multi-tiered framework correctly identified all anomalous data points.

The choice between using a balanced or unbalanced dataset to train the computational techniques depends on the time constraint and on the ability to sample from the entire normal dataset. For instance, an unbalanced dataset is typically larger, and therefore necessitates longer training times for all computational intelligence techniques.

C. *Online Learning*

With offline learning, the computational intelligence techniques are trained only once and then applied. Therefore, no training time is spent during the problem solving process. However, this type of learning constrains the use of the framework in dynamic problem environments. Online learning, on the other hand, is more suitable to dynamic problems, where the training process is carried out using recently obtained data points, making the computational intelligence techniques more adapted to the current problem solving stage. With the use of parallel and/or distributed computing, the training process can be speeded up, allowing for near-real time problem solving capability.

The results on the online learning are shown in Fig. 5, Fig. 6 and Fig. 7. These results are obtained using an artificial neural network and a support vector machine. The fluctuations in the results presented in these figures can be explained by the fact that the chosen computational intelligence techniques are retrained every time three (or more) unseen anomalies are identified.

Similar to the case of offline training, two types of datasets are used in this set of experiments – balanced and unbalanced. Fig. 5 compares the percentages of correctly identified instances, and shows that the performance of the computational intelligence techniques is greatly influenced by the selection of data points used in the training process (for a balanced dataset).

However, with a smaller number of training instances, especially at the later training iteration, the training time on the balanced dataset is much quicker when compared to the training time with the unbalanced dataset.

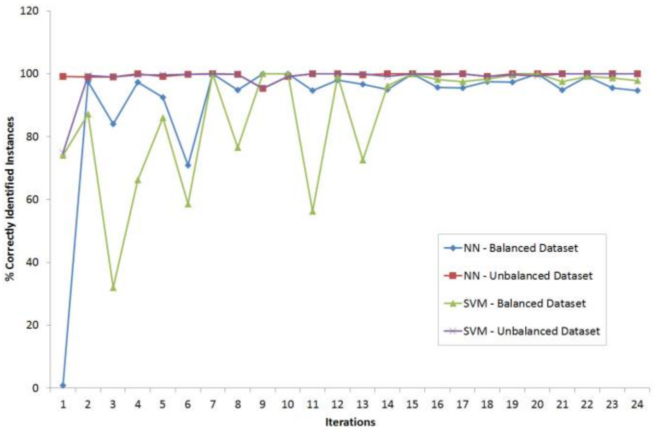

Fig. 5. The comparisons of correctly identified instances between aritifical neural network (NN) and support vector mahcine (SVM)

Fig. 6 compares the number of false negatives and false positives when an unbalanced dataset is used for training the artificial neural network and support vector machine respectively. The number of false negatives and false positives are similar for both computational intelligence techniques; however, a larger percentage of false positives after the first training iteration can be observed in the case of using an support vector machine. The percentages of false negatives are larger than the number of false positives when an unbalanced dataset is used for training both techniques.

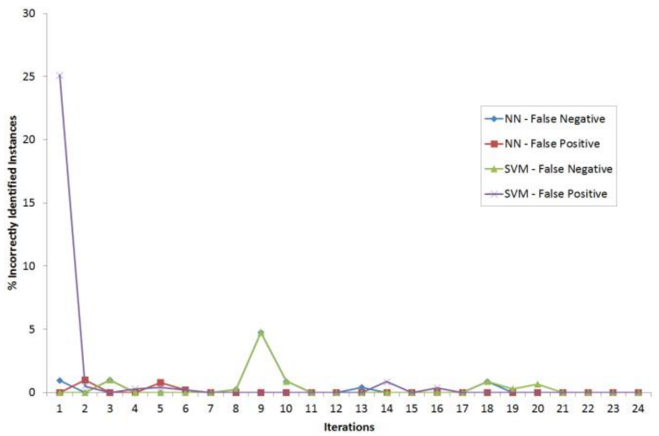

Fig. 6. The comparisons of incorrectly identified instances between the number of false negatives and false positives on an unbalanced dataset using aritifical neural network (ANN) and support vector mahcine (SVM)

Fig. 7 compares the number of false negatives and false positives when a balanced dataset is used to train the artificial neural network and support vector machine respectively. Both techniques correctly identified all anomalous data points;

however, the numbers of false positives fluctuated from one training iteration to the next for both CI techniques. When compared to the results in Fig. 6, the unbalanced dataset performs better in terms of the number of false positives while the balanced dataset performs better in terms of the number of false negatives.

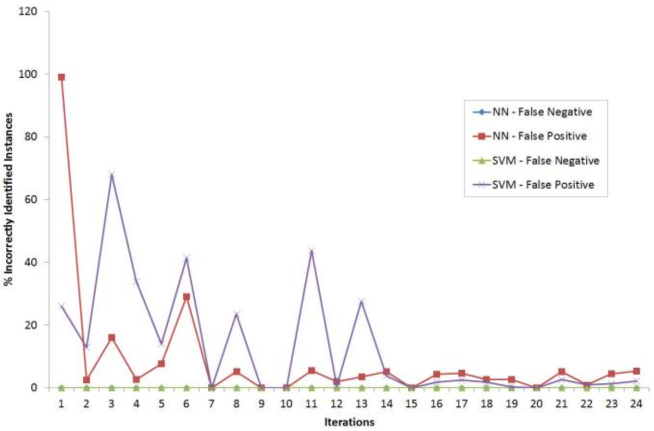

Fig. 7. The comparisons of incorrectly identified instances between the number of false negatives and false positives on a balanced dataset using aritifical neural network (ANN) and support vector machine (SVM)

Table 2 illustrates the effect when different numbers of newly identified anomalous instances are used as parameters for the framework to perform a retraining process for the chosen computational techniques. The results are obtained by applying an artificial neural network and support vector machine trained on both balanced and unbalanced datasets. In this experiment, the computational intelligence techniques are retrained when 3, 5, 10, 15 and 20 new anomalous data points are identified.

TABLE II. THE PERCENTAGES OF CORRECTLY IDENTFIED INSTANCES USING DIFFERENT NUMBER OF NEWLY IDENTIFIED ANOMALOUS DATA POINTS BEFORE RETRAINING

	3	5	10	15	20
NN (Balanced)	91.34	91.82	97.15	98.09	97.97
SVM (Balanced)	87.14	87.21	85.79	86.14	90.71
NN (Unbalanced)	99.54	99.71	99.54	99.82	99.88
SVM (Unbalanced)	98.48	99.64	99.58	99.68	99.72

The percentage of correctly identified instances generally increases when a larger number of newly identified anomalous data points are identified before a retraining process is carried out. The percentage of correctly identified instances is also dependent on whether a balanced or unbalanced training set is used. The proposed framework performs better when the unbalanced training dataset is used. However, the training process in the latter case is longer because of the larger training dataset.

D. Meta-learning

A major drawback of many computational intelligence techniques is the process of obtaining optimal parameters. Generally, the parameters are set empirically by combining the use of a human expert and conducting tuning experiments in order to obtain the optimal setting.

In the multi-tiered framework, this process is assisted by a meta-learning procedure that uses genetic algorithms (GA). The inputs from the sensors at the context-acquisition layer can be analysed in terms of the presence of anomalous data using three different mathematical functions: moving average (2), instantaneous change (3) and standard deviation (4):

$$MA(W)_i = |\, X_i - \frac{1}{W} \sum_{j=0}^{W-1} X_{i-j} \,|. \quad (2)$$

$$\Delta(W)_i = |\, X_i - X_{i-W} \,|. \quad (3)$$

$$\sigma(W)_i = \sqrt{\frac{1}{W} \sum_{j=0}^{W-1} (X_{i-j} - \mu_0)^2}\,,\ \mu_0 = \frac{1}{W} \sum_{j=0}^{W-1} X_{i-j}. \quad (4)$$

Each of these mathematical functions also depend on the window size W. Ten different values of this parameter were used in the meta-learning experiment, these were 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50.

The meta-learning process in the framework has been implemented by applying a genetic algorithm with fixed-size chromosomes that encoded the input processing mathematical function (2) – (4) and the corresponding window size W as chromosome alleles. Other genetic algorithm parameters are: the crossover rate (50%), mutation rate (10%), the population size is 10, and the percentage of population allowed to mate is 24%.

The fitness of each chromosome is proportional to the number of incorrectly identified instances in the input data stream (i.e. the total number of false negatives and false positives) when a particular combination of functions (2) – (4) with the corresponding window sizes W is used to process the sensor inputs and to train the computational intelligence techniques – artificial neural network in our experiments.

The input nodes for the artificial neural network consist of current input sensor values and the ones processed using the mathematical functions encoded in the chromosomes. Fig. 8 illustrates the square root of the mean, the median and the minimum fitness of the genetic algorithm's population over time. The first performance measure is used to reflect the worst scenarios when the artificial neural network misinterpreted a large proportion of input data.

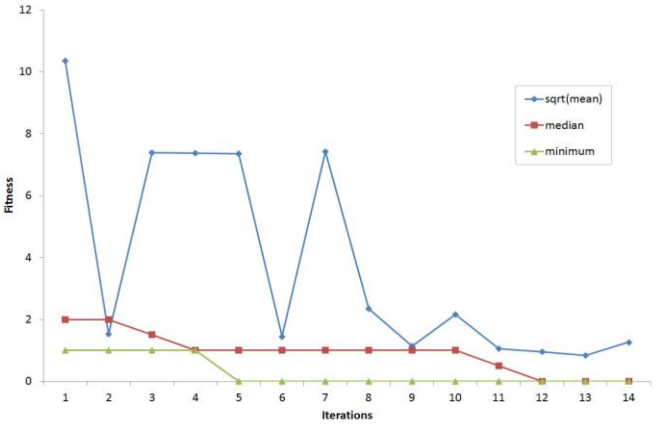

Fig. 8. The performance of a genetic algorithm meta-learning procedure

As can be seen from Fig. 8, the fitness of the population improves over time. Due to the simplicity of the fitness function, it appears that different input processing functions produce similar overall fitness values. A more complex fitness function can be used to include other factors affecting the performance, such as input processing and computational intelligence technique training times. The meta-learning procedure can be made more sophisticated so that it can be applied to other computational intelligence techniques (SVM, BAYES), or even to enable selection of an optimal computational intelligence technique most suitable for solving the problem at hand.

V. Conclusions

The developed adaptive multi-tiered framework for automated ad-hoc navigations of the robot and the controls of its sensors has been developed and extensively tested. Adaptability and autonomy of the framework are achieved through the combined use of statistical analysis and computational intelligence techniques. By selectively choosing a different number of inputs at the three phases of processing data (i.e. the identification, classification and prediction), the effects of various anomalies in the input stream can be made more effective.

The experiments conducted on several datasets have demonstrated that reasonable performance is achieved in terms of accuracy of data processing and its speed. The versatility of the proposed methodology is demonstrated by successful applications of the suggested framework to tracking a wall-following robot, to surveilling traffic [3], to environment monitoring in smart homes [4], and to controlling the level of electromagnetic interference in motor vehicles [10]. Moreover, the performance of the proposed framework operating in the online learning mode also looks promising, even with a small number of data points used in the training process.

In the case of online learning, it may be useful to retain previously obtained parameters and settings of the computational intelligence nodes. However, this would imply that additional time is required for comparing previously obtained settings with the current ones. If the optimal settings are required to achieve the best performance and the processing time is not a crucial factor, then the comparisons would allow for fewer fluctuations in terms of the number of correctly identified instances of anomalous data, especially in the case when a balanced dataset is used for training. In a dynamic problem environment, these optimal settings are seen as less important however.

In data-driven intelligent systems, input variables, the type of statistical measures used for pre-processing the input data, and the parameters of various computational intelligence techniques are usually obtained empirically. Another contribution of the work presented in this paper is the adaptation of a meta-learning procedure that is used to semi-automatically tune the set of input variables, the techniques to pre-process input data, and the salient features of the computational intelligence algorithms capable of identifying data patterns of interests that are used by the context-processing and context-selection layers of the multi-tiered framework. As was demonstrated in the paper, the meta-learning procedure shows some potential in autonomously guiding the adaptive selection of the most effective salient features and parameter settings for the computational intelligence techniques.

The data-driven evolving choices of the genetic algorithm and autonomous system has the generality to be applied across a wide range of problem domains requiring processing, analysis and interpretation of data obtained from heterogeneous resources. Further investigations will be directed towards extending the functionality of the system, improving its efficiency, and enhancing its scalability.

References

[1] Veres, S. M., Lincoln, N. K., & Molnar, L. (2011). Control engineering of autonomous cognitive vehicles-a practical tutorial.

[2] Z. Khan, A. B. M. Shawkat Ali, and Z. Riaz (Eds.), Computational Intelligence for Decision Support in Cyber-Physical Systems. Springer, 2014.

[3] P. Rattadilok, and A. Petrovski, “Inferential measurements for situation awareness: Enhancing traffic surveillance by machine learning,” in Proc. Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2013, pp. 93-98.

[4] P. Rattadilok, A. Petrovski, and S. Petrovski, “Anomaly monitoring framework based on intelligent data analysis,” Intelligent Data Engineering and Automated Learning (IDEAL 2013), Lecture Notes in Computer Science, Vol. 8206, 2013, pp. 134-141.

[5] K. J. Park, R. Zheng, and X. Liu, “Cyber-physical systems: Milestones and research challenges,” Computer Communications, Vol. 36(1), December 2012, pp.1-7.

[6] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Computing Surveys, Vol. 41(3), September 2009, pp. 1-72.

[7] Encog Library, http://www.heatonresearch.com/encog.

[8] UCI repository, http://archive.ics.uci.edu/ml/datasets/Wall-Following +Robot+Navigation+Data.

[9] WEKA, http://www.cs.waikato.ac.nz/ml/weka/.

[10] S. Petrovski, F. Bouchet, and A. Petrovski, “Data-driven Modelling of Electromagnetic Interferences in Motor Vehicles Using Intelligent System Approaches,” in Proc. IEEE Symposium on Innovations in Intelligent Systems and Applications, (INISTA), 2013, pp. 1-7.