

# The Genomics of Industrial Process Through the Qualia of Markovian Behavior

Morad Danishvar<sup>1</sup>, Alireza Mousavi<sup>2</sup>, *Senior Member, IEEE*, and Sebelan Danishvar

**Abstract**—A technique for registering and relating events that cause an observable and definable system state is proposed. Discrete events of system-state transfer are expressed by event tracking and clustering in the form of contiguous quanta of data. This approach is capable of describing typical processes in industrial systems in a chain of codes that contain system input/output parameters. The constituent nodes of the Markovian Processes chain form a series akin to genes in the deoxyribonucleic acid, repeatable and predictable. The process genes are the quanta of information that aligns to represent a chain of activities (process). They describe the causal links between occurring events forming a pattern (pathway) that leads to a well-specified output (e.g., a product with a defect or otherwise). The creation of process genomics requires the knowledge of system observed or latent parameters (state) as well as the state change at specified time intervals (discretization). The process genomics theory is tested in an industrial case study for quality assessment and control of glue dispensing in micro-semiconductor manufacturing. The resulting definitions of the system state and interrelationship of control parameters contribute to the development of the *process genes*. The outcome of the gene alignment is the geometric interpretation of the glue droplet formation. A predicted or observed droplet within the production tolerance leads to a nondefective product. The principle of creating production genomics is to find and rectify the defect-causing genes or to disrupt the sequences that lead to producing defective products, leading to a zero-defect manufacturing process.

**Index Terms**—Data driven, dispensing technology, machine learning, manufacturing, Markovian process, prediction, quality, real time.

## I. INTRODUCTION

SYSTEMS equipped with sensory inputs should be able to recognize and predict temporal sequences of events. A sequence defines as an ordered series of events. Depending on the type of events forms sequences, discrete, continuous, or binary sequences are some examples of different kinds of sequences. In real-life applications, we face these types of sequences that can be categorized between the essential forms of sequences.

Manuscript received 24 January 2022; accepted 5 February 2022. Date of publication 21 February 2022; date of current version 17 October 2022. This work was supported by the European Union’s Horizon 2020 Research and Innovation Program under Grant 723906. This article was recommended by Associate Editor L. Sheremetov. (*Corresponding author: Morad Danishvar.*)

The authors are with the College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, U.K. (e-mail: morad.danishvar@brunel.ac.uk; alireza.mousavi@brunel.ac.uk; sebelan.danishvar@brunel.ac.uk).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSMC.2022.3150398>.

Digital Object Identifier 10.1109/TSMC.2022.3150398

Sequential behavior analysis is a key element of human reasoning, complex problem solving and decision making. Especially, sequence learning is a major module of learning domains, such as natural language processing, machine learning, adaptive control, temporal prediction, financial engineering, genome sequencing, and so on [1]. The process of this temporal information is also one of the fundamental aspects of human intellectual ability and his desire to replicate this in industrial automation and computing.

A sequential phenomenon of events can be expressed in the form of an ordered or random list of symbols or numbers. The knowledge about the list of events can be enhanced by, for example, registering the time they occur (timestamp). Event sequencing is, therefore, registering as an observed phenomenon at specified time intervals. The reported sensory data in such time series can be compared with earlier data, and state changes or nonchange be determined. Equipped with such preliminary data and logical inference, there is a possibility to go beyond, and that is event sequence prediction (ESP). ESPs consist of predicting the occurrence of the next symbol(s) in a sequence based on the previously observed symbols. If such an occurrence has not been observed previously, then a new event is registered.

It is safe to declare that the logic and chain of reasoning (theorem) of the proposed genomics of industrial process (GIP) is closely related to temporal sequence learning (TSL) terminology. Imitating the human brain, the GIP learning process exploits the temporal sequence and its rationale to explain and project the state of systems in time-space.

Sequence prediction is classified as an application of sequential data and attempts to predict elements of a sequence based on the preceding elements in

$$S_i, S_{i+1}, S_{i+2}, S_j \rightarrow S_{j+1} \\ (S_i, S_{i+1}, S_{i+2}, S_j, \text{ is given and } 1 \leq i \leq j < \infty) \quad (1)$$

$S_{j+1}$  is the definitive prediction output. When  $i = 1$ , we predict based on all the previously occurred elements of the sequence. When  $i = j$ , we predict based on just the next element (e.g., Markovian processes).

Sequence classification is another example of sequence prediction. It includes the prediction of a class label for a specified input sequence. For example, deoxyribonucleic acid (DNA) sequence classification falls into this category. Given a DNA sequence of A, C, G, T (where each letter represents one of the four basic constituent molecules collectively known as nucleotides: 1) cytosine (C); 2) guanine (G); 3) adenine (A);

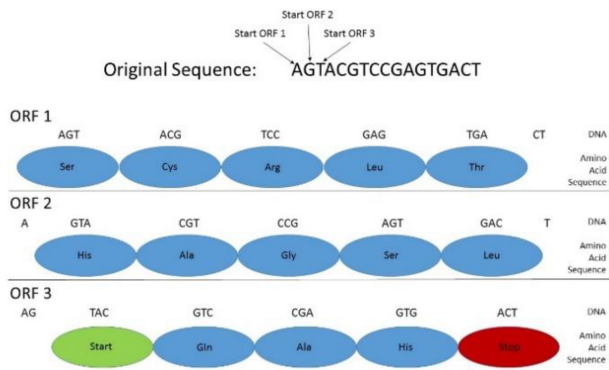


Fig. 1. Example of gene prediction (borrowed from [4]).

and 4) thymine (T)) values, predict whether the sequence codes for a coding or noncoding region [2], [3]. A genomic DNA segment, such as AGTACGTCCGATGACT, is a string of amino acids without any temporal connotation attached to their order in the sequence. Fig. 1 shows an example of how genomic sequencing can be used for gene prediction by open reading frames (ORFs). Gene prediction is the process of determining where a coding gene might be in a genomic sequence. The length of functional proteins is from where the DNA transcription begins (i.e., start codon), to where it ends (stop codon). Therefore, a functional protein is searchable by codons start and stop points in a DNA sequence. This is important in gene prediction because it can reveal where coding genes are in an entire genomic sequence. In this example, a functional protein can be discovered using ORF3 because it begins with a start codon, has multiple amino acids, and then ends with a stop codon, all within the same reading frame.

There are two steps to make a sequence prediction which are described as follows.

- Step 1: Training sequences are the task of training of a sequence prediction model by using some previously seen sequences. For instance, a sequence prediction model can be trained for machine operational and control sensors data for the prediction of a specific type of defect in the manufacturing plant.
- Step 2: Use the trained sequence prediction model to predict the new sequences (i.e., predict the next element of a new sequence). For example, using the trained defect prediction model to predict the upcoming defects.

Interpreting and relating the sequence of events that occur in a production process allows formulating the causal relationship between events. The expectation of this activity is to create a genetic construction of the process and is the main contribution of this research work. Such a construction is analogous to medicine (i.e., *good gene* causes healthy outcomes, and *bad gene* causes illness). It provides a perspective to predict the outcome of a process (e.g., end product Defect/Pass) based on the genetic constellation of the process, as the contiguous, i.e., events chain up during the process life cycle. The genetic register of the process will be saved in a gene/DNA of the process library, where the “good genes” (optimum solution creating events) and “bad genes” (fail conducting events)

and the sequence of their occurrences will be registered and used for optimization purposes by encouraging good genes and eliminating bad genes. In the latter case, by adjusting machines, material, logic, etc., to prevent/avoid the occurrence of bad genes (e.g., defect inducing events).

For ease of understanding and to explain event-based process sequencing, the proposed GIP concept borrows some of the classical terms and descriptors of genetics and genome sequencing from biology science.

The proposed GIP registering technique is tested and validated through an industrial case study for quality assessment in printed circuit board (PCB) fabrication and assembly manufacturing. The creation of production genomics helps in the prediction of defects and prevention through rectification of the defect, causing genes, leading to a zero-defect manufacturing process. To the best of our knowledge, such an approach to in-process Quality Control and Management is a step change compared with classical online–offline and post-process quality analysis.

In the following sections, a review of the most relevant sequence learning methods available in the literature is presented, followed by a detailed description of the proposed method and its application in an industrial case study. For performance comparison purpose, the same experiment is conducted by GIP and some other neural network (NN) methods for defect prediction. The comparison helps the authors to explain the merits and applicability of the proposed method for problem solving in real-world industrial applications.

## II. RELATED WORKS

Sequence learning’s problems complexity and diversity are so varying that a single approach could be able to suffice to master the field. In the literature, there are many different applications for sequential learning, from the prediction scheme in navigation [5] to pattern recognition and prediction in [6]–[8].

Depending on the nature of the system and type of data, several methods have been proposed and applied. In this section, several relevant methods to the proposed technique will be reviewed.

Machine learning techniques examine the pattern of seen data and generate rules to discover recurring patterns. Machine learning algorithms learn from a sequence of events and “experiences” with respect to a class of tasks [9]. The concept of machine learning is, therefore, the study of approaches and methods which can be applied to generalization and learning observed/seen instances pattern and construct rules in order to make a prediction when faced with new instances [10]–[13].

A taxonomy of machine learning methods is presented in Fig. 2 [10]. Different machine learning methods have different approaches to learn sequences patterns. In the continue, some methods which use TSL will be explained briefly.

Hidden Markov models (HMM) is a method deployed for sequence learning (including both generation and recognition). This model learns fundamental state transition probability distributions from observed data. The sequence generation can apparently be managed by this type of model. A comparison

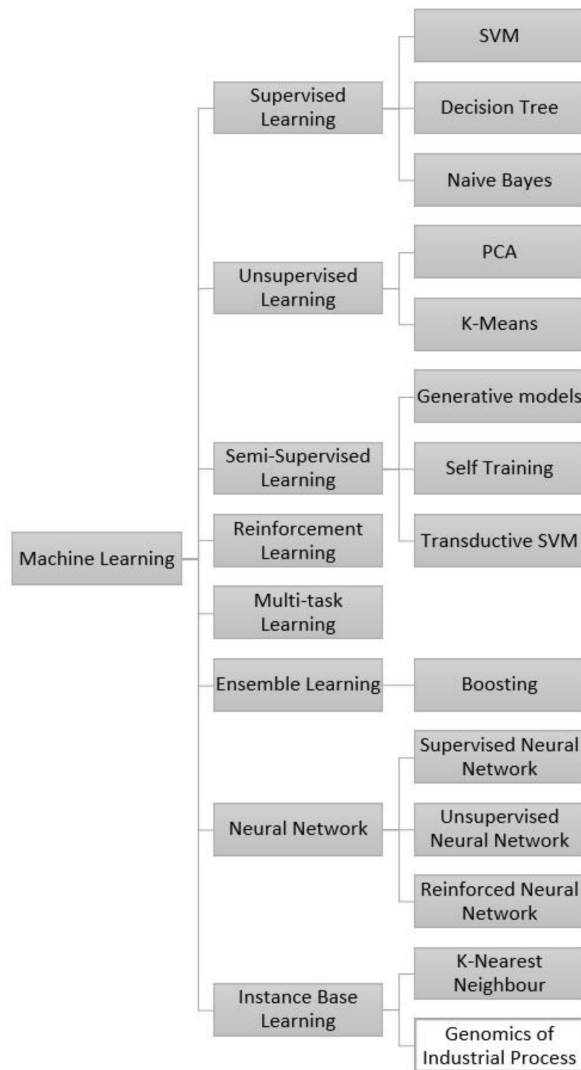


Fig. 2. Types of learning.

of supervised and unsupervised learning approaches of HMMs has been reviewed in [14]. After a theoretical comparison of both methods, a controlled experiment compares results obtained. The outcomes present that supervised learning methods performance is poor because they impose binding conditions in terms of data labeling, involves applicants' biases, calculate unreliable results due to the absence of constructs an efficient Maps with higher performance, and fewer applicants intervene. Temporal-Difference [15] (used mainly in reinforcement learning) and NN learning methods [16] are other examples of sequential learning of temporal patterns.

A time series represents a sequence of acquired values that are measured through time span. Such a series includes an ordered sequence of observations of a finite-sequence length, which are often taken through time span or space. For decades, time series have been applied in prediction and forecasting applications and theories. However, mostly, they rely on mathematical equations, simulation and/or learning techniques to represent the evolution of time series data. The body of literature in time series is devoted to time-series classification applications in machine learning and a wide range of industries [17]–[19].

One popular TSL-based machine-learning technique is instance-based learning (IBL) [12]. In this technique, an instance dictionary stores the set of representative instances. New unseen instances would be classified according to their relationship to previously seen stored instances.  $k$ -NN is an example of the instance selection methods which were developed based on IBL algorithms.  $k$ -NN is a typical example of classification methods, in which a new instance is assigned the label of many of the  $k$  dictionary instances according to domain-specific “closest” or “similarity” measurement. In temporal-sequence domains, the similarity or closest measure is often taken to be the Euclidean distance [22]. This classification method is between top-performing methods. However, the complexity of IBS increases by growing the volume of data. A distinctive problem in  $k$ -NN is deciding which instances should be stored for generalization. Storing too many instances require large memory space and slow computation execution speed [23].

An evolutionary instance selection algorithm for IBL introduced by de Haro-García *et al.* [24]. Although evolutionary algorithms are efficient in performance, a pitfall of them is the essence of storing all training instances subsets in memory space for large datasets, which might also impact the efficiency of the testing task. A solution to this problem introduced in [25] with an instance selection method that removes redundant and noisy instances. Gong *et al.* [26] provided a survey of existing techniques used to reduce storage requirements in IBL algorithms, including different reduction and/or reconstruction algorithms of the training set. However, such storage reduction advantage usually becomes less interesting when we know it compensates with accuracy reduction. One of the recurrent problems with these instance selection methods is that although significant reductions in storage requirements are obtained, this reduction often comes at the cost of degraded accuracy.

IBL algorithms have a wide range of applications, from stock market prediction [27] to anomaly (Outlier) detection [28]. In anomaly-detection tasks, the anomalies can be formulated from learning of behaviors of the system in terms of temporal sequences of data. Chandola *et al.* [29] presented an approach to cast the anomaly-detection task in an IBL framework. Their approach is to transform the temporal unordered sequence of observations into metric time-spaces via a similarity/closest measure that includes their intraattribute dependencies.

With the growing availability of streaming online/real-time data, there is a significant demand for online/real-time sequence learning algorithms. With a focus on the TSL terminology and IBL methods' performance and storage requirement problems, in the next section, an online/real-time sequence learning algorithm named ESP technique with a novel storage reduction method is introduced. This sequence learning algorithm is more efficient in that it requires much smaller computing and storage resources. The proposed ESP technique learns the event of complex sequencing in real time while existing types of learning (such as statistical and machine learning) methods are not well suited to solve such real and practical needs and desires from industrial applications.

### III. SEQUENTIAL EVENT PREDICTION—THEORY OF EVENT-BASE GENOMICS OF INDUSTRIAL PROCESS

The Microelectronic fabrication and assembly process at times creates up to 50% defective products that need to be thrown away. Classical quality control techniques mainly rely on post-process quality assessment and statistical process control methods. Experts will then try to relate the patterns of quality loss to machine and material states at the time of production. Such techniques have limited impact and have proven obsolete with modern manufacturing processes. The intention here is to develop an accurate and applicable quality assessment and process correction system that could adjust the production system so that during the process, potentials of quality losses are identified in real time, the system is alerted, and corrective measures taken, leading to the optimal prize of zero-defect manufacturing (EU H2020 FoF Research and Innovation Program under Grant 723906).

As reviewed in the related research and literature section, all current data-driven and learning algorithms that could meet such complex demand were not available. All learning-based methods require training and large sets of data that neither exists about the raw material feed, the state of the machine and the type of defects generated. Furthermore, no formal and verifiable data/knowledge exists of the correlations between systems parameters that relate to quality loss (defect analytics).

The challenge was to develop and upgrade the real-time data acquisition on the production process and find a method of automatically interlinking the causal relationships between raw material state as well as machine state and capability of meeting product design specifications. This is especially timely since, in industrial applications, quick understanding of the state of the system (input parameters of system) and taking action means savings and improvement in quality, productivity, energy efficiency and sustainability performance (latent system outputs). Therefore, the closer the analysis and action is to the live operations, the more useful they will be in practice. Furthermore, the application should be easy to implement and follow by industrial controllers, operators, and decision-makers. The proposed method introduces a general framework for using the event-clustering technique [34] in ESP. It has been named as a theory of event-based genealogy of process since this prediction method borrows the terminology used in genetics (in general terms) and, similarly, labels process events. Akin to DNA chains, the string of events is symbolized, representing manufacturing process causal relationships and contiguous occurrences.

Analogous to the formation of polypeptides chains that manifest themselves as functions of life at the molecular level, each chain of signals in an industrial system registers function in a process. The chains present the linear sequence of events where a specific constellation can be interpreted as the building blocks of a process. For example, the chain of events that forms the DNA of process that successfully produces a “good product” or otherwise. The sequence of events that normally leave a traceable signature produces a registrable gene. Some constellations of genes (DNA) will lead to “good” products, and some lead to defects. Identifying such defect generating

events will eventually lead to detection and prediction of the causes of a defect in a process (output). By identifying the defect generating genes in the process, methods for detecting them and countering (control reactions) or remedial actions (change of pathway) will lead to reduction and elimination of defect and waste in industrial systems. In this article, an example of manufacturing, where the genome of defect type  $x$  is presented as a sequence (or chain) of events (genes alphabetically labeled) that shifts state of the system to the generation of defect  $x$  is discussed. Defect  $x$ , for example, can be defined as a combination of repeatable events labeled A–D (alphanumerically assigned), and defect  $x$  would happen if a sequence of ABBCD occurs. A, B... are representative of distinguished system states, which will be explained in detail in the next sections. Noteworthy, the application of this method is not limited to quality control and could be used for other purposes such as process control (stabilization) application. That is, instead of the registration of bad genes that lead to defects, good genes which make the system stable in an optimized or desired point through registered scenarios/sequences can be registered.

What distinguishes the proposed ESP techniques from other sequence learning techniques is its simplicity and the speed at which it extracts and learns all existing sequential learning as well as its storage reduction technique and then processes the necessary information in near real time. There is no reliance on a set of predefined rules, such as good or bad genes or time-consuming investigation on the patterns. More importantly, unlike heuristic methods, the proposed technique does not rely on any prejudgment of the data relevancy, that is, normally a characteristic of expert interference and in that respect, it is an unbiased method.

#### A. Theorem of GIP

The assumptions and basic parameters of the proposed EventTracker and Event clustering (EventiC), such as discrete event system (DES), tigger data (TD) and event data (ED), trigger and event threshold (TT, ET) are presented in [30] and [31]. Further parameters of the proposed sequencing method are as the following.

- 1) *Sequence*: A Sequence is expressed as the value or state of the actual I/O data at a given time instance. These data can be expressed in a binary, integer, or decimal format. For instance

$$(I_1, I_2, \dots, I_m) f_t(O_1, \dots, O_n). \quad (2)$$

$f_t(O_1, \dots, O_n)$  represents the value of outputs at a specified time instance.

- 2) *GIP*: A GIP is the sequence of events that lead to a definitive output(s). The definitive outputs are observable/measurable states (e.g., defect type  $x$ ) called a genome (process) of defect type  $x$  formation.
- 3) *Length of GIP*: The length of GIP is the number of elements (i.e., genes) in a sequence. At present, determining the length of genomes are specified by the definition of process (processing steps) or expert knowledge. Trial and error in the period of learning can also be an alternative.



**B. Sequence Database**

All labeled genomes are stored in a database called a lookup table. These genomes are labeled, timestamped, and contain a definition. This database is continuously updated with new events and repeated events; thus, the gene pool will be identified after a period of observing the system. For each type of industry, such genes become unique or shared. Thus, the family trees of processes, machines, tools, and networks could emerge.

**C. Sequence-Generating Rules, Genealogy of Process**

A set of rules and protocols govern the constraints and objectives of the system. For example, a definitive output or input boundaries and constraints could be defined by these rules and with respect to the nature of the system studied. A genealogy of process will emerge. With the advent of machine networks and large-scale monitoring and integrated systems, the necessary data for this process will exist. For example, a specific machine that produces micro semiconductors will have a pedigree with shared genes with the same brand, and more importantly, with other brands, such interrelationships will build the family trees, where behaviors and patterns can be extracted for better control logic. New generations of machines/processes can be created by eliminating fault generating machines leading to evolution and optimization of the design.

**D. Sample Scan Rate**

Sample scan rate is a specified time interval, where the duration can potentially range from microseconds (high frequency) to hours (low frequency) and is chosen by the system expert based on the nature of the application. For example, applications such as safety systems require shorter scan intervals between events, whilst other applications, such as wastewater plant process require longer intervals [31].

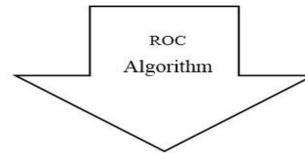
**E. ESP Algorithm**

The proposed ESP ranks and groups the relationship between system input and output parameters. A combination of event tracking [31] and rank-order clustering (ROC) [30] groups and assigns weights to the system input against system outputs. The array of inputs and their weight vis-à-vis system outputs are used in generating the event sequences genomes and their likelihood of occurrence. In the following, a step-by-step implementation of the method is presented.

1) *Implementation of the EventiC Algorithm:* The ESP algorithm begins with dimensionality reduction through an ROC technique. EventiC sensitivity analysis helps to find a correlation between inputs and output and then calculate the weight of impact of each input on the output parameters. It runs ROC (See in Fig. 3) to identify the group of most relevant inputs and outputs.

2) *Definition of Rules and Definitive Output(s):* The system inputs boundaries and definitive output(s) are defined by a set of rules. The system inputs boundary and constraint help the outliers, such as noise removed from the dataset. Moreover, the

TDs/EDs	1	2	3	4	5
1	0	1	1	0	1
2	0	1	1	0	1
3	0	1	1	0	1
4	1	0	0	1	0
5	0	1	1	0	1
6	1	0	0	1	0
7	1	0	0	1	0
8	0	1	1	0	1



TDs/EDs	1	4	2	3	5
4	1	1	0	0	0
6	1	1	0	0	0
7	1	1	0	0	0
1	0	0	1	1	1
2	0	0	1	1	1
3	0	0	1	1	1
5	0	0	1	1	1
8	0	0	1	1	1

Fig. 3. Input triggered/ED incidence matrix before (up) and after (down) implementation of the ROC algorithm.

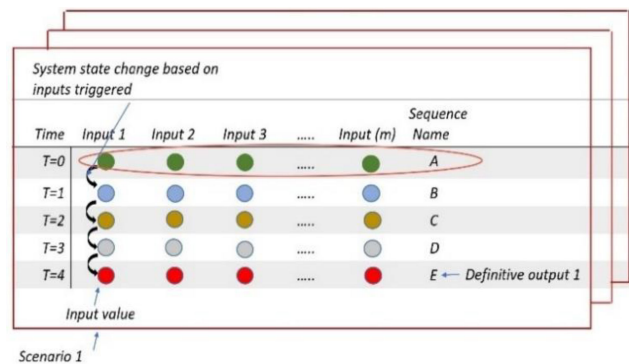


Fig. 4. Genome labeling and sequential differentiation example.

rules set includes the definitive system outputs. The definitive outputs are the range of outputs we are searching for as the objective of our prediction. For example, if the aim is to predict defects by the ESP algorithm, the definitive system output(s) is the specific type of defects and their properties.

3) *Genome Labeling and Sequential Event Differentiation:* The next step in the ESP algorithm is the labeling of the events. According to the definitions, states (e.g., defect) detection is conducted. Then, based on the chosen genome length, all contiguous sequences of events prior to the defect state are labeled.

An example will demonstrate how patterns are discovered patterns in event sequences. Fig. 4 illustrates the labeling and sequence differentiation process. Here, the system observes the definitive output(s) at  $T = 4$ . Then, the value of relevant sequences will be read and labeled in the following time sequences:  $t_0, t_1, t_2, t_3, t_4$ .

The first element in the sequence is registered at  $t_0$  and labeled as “A.” According to the EventiC output and TD/ED, the second sequence labeled could be a new state differentiable

to the prior state, only if the system state's difference passes the threshold, then a new label is assigned to the new state as "B" this is continued until the moment of state E which is the occurrence of the defect.

Subsequently, the formation of events from  $t_0 \cdots t_4$ , five consecutive events form the genome strand "ABCDE." Note that in this example, only four prior events are considered sufficient to form the genome of defect formation.

4) *Training and Storing the Genomes:* At the end of the training period, all trained genomes/scenarios will be stored in the sequence database as a lookup table. In real-time prediction, these stored genomes will be used for the prediction of the system trained definitive outputs(s) and declaring the likelihood of occurrence. In the above example, if during prediction, sequence A is detected, the likelihood of defect will be 20% (1 out of five possible events. Consequently, if "A" → "B" → "C" → "D" occurred in the real-time prediction, more likely the next sequence is "E" unless proven otherwise. However, if state E did not occur, and "F" occurs, the algorithm has detected two different genomes for a definitive output, and the genome length must be extended to six to generate new labels and genome.

#### F. Level of Confidence for ESP Results

The length of a GIP needs to be extended if the GIP prediction leads to two different definitive outputs. The length would be extended until the GIP meet a unique definitive output. A case study is presented in the following section to further explain the GIP implementation in a real industrial case. Here, every step of the process is registered in the form of a chain of events coded in the form of a sequence of genomes representing the DNA of process scenarios that lead to good or "defect" outcomes.

### IV. DNA OF MICRO SEMICONDUCTOR MANUFACTURING PROCESS

A defect-prone process of a PCB fabrication and assembly is presented. The assembly process requires consists of dispensing conductive glue on a Liquid Crystal Polymer (LCP) substrate with tight design specifications; the microelectronic components need to be precisely positioned on the wafer using a machine. The process involves the placement and adhesion of die/components the size of  $800\mu\text{m} \times 900\mu\text{m}$ ) with conductive paste into a laser-cut cavity. The assembly process is undertaken on a mother panel containing 18 individual circuits each circuit containing 20 components. Prior to the deployment of the proposed GIP, the placement machine did not have any online optical inspection capability and no method to automatically correct the amount of glue dispensed in real time into the cavity. Consequently, a visual volumetric inspection of the assemblies (post-process quality control) took place after batches of products were made; subsequently, an analysis of machine and raw material states were made, and adjustments were made to the machine retrospectively at a scheduled maintenance period. Note that the old practice produced a significant number of defects between 5% and 50%. As the

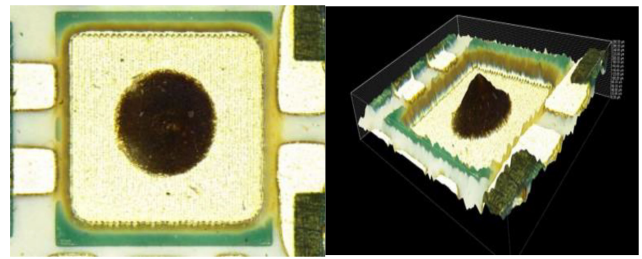


Fig. 5. Single glue dot 2-D (left) and 3-D (right) images.

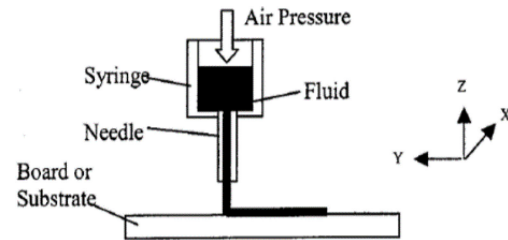


Fig. 6. Time-pressure dispensing schematic [36].

design becomes more complex, the defects increase exponentially. The glue dot dispensed volumes require to be controlled; otherwise, it might lead to some severe defects. The possible defects are as the following.

- 1) Excess of glue—leak up the excess of glue to the side of the die causing solder shorts—action: reject to scrap.
- 2) Insufficient glue—it causes component adhesion issues—action: rework/scrap.

The actual single glue dot is shown in Fig. 5. There is a need to be able to improve this process and ideally control the amount of glue being dispensed into the cavity automatically and during the production process. As a first step, introducing an online automated optical inspection machine would help to provide the necessary data for analysis and trend learning.

#### A. Rheological Behavior of Glue

To define output system parameters (observational and modeled), characterization of the time-dependent rheological behavior of glue is conducted in [32]. With air compressibility and liquid inertia consideration, an inferential model of the dynamics of the flow rate of the dispensed liquid is sensitive to the air volume in the syringe (i.e., the definitive output). An experiment emulating the machine glue dispensing was conducted in [33]. Given various glue characteristics, needle conditions, operational and environmental conditions, the experiments were repeated. Various control mechanisms were deployed to capture and reduce inconsistency (e.g., defining the range of glue dispensing patterns; thus, the operational ideal (good drops) and nonideal drops (defective) were extracted.

A schematic diagram of the system is shown in Fig. 6. This is a schematic of a conventional time-pressure dispensing process. An air supply is utilized to give pressurized air in a blend with a valve to control the term of the pressurized air. Through a transmission line, pressurized air is connected to a syringe. In addition, the syringe pushes the liquid out of the needle. Once the liquid is released from the needle, it

drops onto a board, then it streams or spreads on the board to the point that a balanced profile is shaped. The needle is operational typically for 30 000 to 60 000 and lasts 14 days.

The relationship between the amounts of dispensed glue, glue level in the syringe, applied pressure, glue temperature and machine depression (the needle age) have been taken into consideration. This model shows a high sensitivity of the amount of glue dispensed to applied pressure and glue temperature, and the model can predict the amount of glue dispensed.

### B. Measuring the Volume of Glue Dot

Optical inspection is a nondestructive technique apply in the electronic industry, including PCB defect detection. A comprehensive literature review in [34] appraises the techniques used to detect defective PCBs. The review also analyzes the inspection algorithms used for detection and interpretation of defects in the electronic components. The algorithms include data preprocessing, feature extraction, and classification.

Excess and insufficient glue on the PCB joints are two types of defects that may occur during the manufacturing and assembly process. Electric tests and human visual inspection tests are common methods of detection and interpretation, but novel automated optical inspection technologies alongside modern inspection algorithms for PCB quality control are gaining impetus [35], [36].

The detection and defect interpretation method in this article deploys a high-resolution laser scanner that scans the areas of interest on a PCB and extracts the geometric information in terms of three-dimensional (3-D) point clouds. Moreover, a regression-net (RNet), a 3-D convolutional NN (3-DCNN) framework, is used to estimate the volume of the glue on the designated die attachment on PCBs [37]. In this study, two well-known deep learning methods of VoxNet [38] and PointNet [39] have been adjusted to the volume estimation with their final classification layer and subnetwork replacement with a fully connected layer NN. In the end, the applied RNet model accurately predicts the volume of glue deposits both before and after die attachment with 92% accuracy. The proposed system uses a custom scanning component that extracts a high-resolution point cloud of a glue deposit or, more interestingly, after die attachment when only a small part of glue is visible from the glue fillet that is formed around each die.

### C. Dispensed Machine Data Parameters and Corresponding Glue Volume

In this experiment, the dispensing machine dispenses 500 glue dots on 10 LCP substrates in 2.5-s intervals. Data about machine state and the ambient conditions during the dispensing were collected from installed operational sensors (temperature sensor (tempSen), air pressure sensor (supplyAPs), vacuum sensor (HouseVa), dispensed voltage (DispensedVolt) and syringe pressure sensor (Syringe Psi)) on the dispensing machine and log to a repository to be stored (Fig. 7). The operation data-sampling rate of the system is set at 500 ms. This implies that the machine state is registered at 500-ms intervals; this allows a 5-event registration

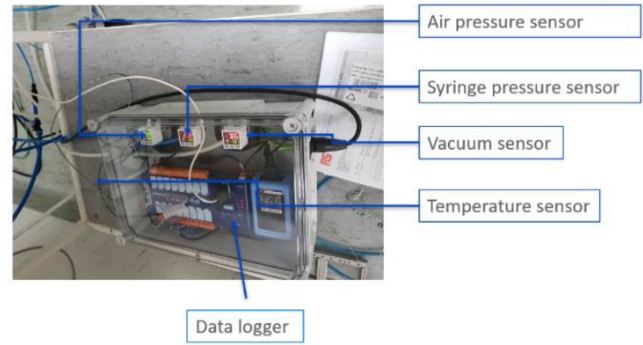


Fig. 7. Sensors and data logger installed on the dispensing machine.

TABLE I  
EVENTIC OUTPUT BETWEEN THE DISPENSING MACHINE  
SENSORS AND GLUE VOLUME

Input/output	Glue Volume
<b>SyringePsi</b>	<b>79%</b>
<b>HouseVa</b>	<b>57%</b>
DispensedVolt	42%
TempSen	38%
SupplyAPsi	29%

prior to the next glue drop. The five contiguous system states could be a good indicator of GIP length. Appendices A and B (Figs. 12 and 13) present all the 500 glue samples of ten LCP glue dots and corresponding machine operational and environmental sensor parameters.

### D. Sequential Event-Modeling Algorithm Training (Scenarios of Process Genomes)

The step-by-step implementation of the proposed ESP is as follows.

1) *Event-Clustering Sensitivity Analysis Algorithm*: The sequential event-modeling algorithm begins with dimensionality reduction through the ROC technique. EventiC [30] helps to find the many-to-many correlation between inputs and output parameters reveals the sensitivity of the glue shape (output) against the machine and ambient state parameters (system input).

The interpretation of the data series is based on triggers and events. Only those fluctuations in the data series that are interpreted as triggers represent state change. False-negative and positive tests were conducted, as explained in [30], and the trigger threshold was set at 5%. It means that any alteration in input sensors (dispense machine operation data) and glue volume, more than 5% has been considered as a new event. The EventiC results are shown in Table I.

These results show Syringe pressure has a high degree of impact on the glue volume while “HouseVa” has a medium impact, and these two sensors should be considered as important sensors in the sequential algorithm. The remaining sensors have a low degree of impact on the glue volume. The low-ranking parameters could be ignored in the next steps (dimension reduction).

2) *Defect Detection*: The second step in the sequential event modeling algorithm is to ascertain the acceptable range for glue volume. The minimum and maximum satisfactory



Dispense number	SECONDS	tempensor (°C)	Supplyair (Psi)	Housevacu (Psi)	Syringeal (Psi)	Dispenses (mv)	Glue Volume	Genome Label
21	51.78	21.993	96.41	-698.85	16.85	3875.6		A
	51.89	21.993	96.41	-695.97	17.56	3875.6		B
	52.00	21.984	96.43	-693.08	18.45	3875.6		C
	52.11	21.975	96.41	-690.20	19.48	3875.6		D
	52.22	21.984	96.44	-689.054	21.38	3875.6	0.31	
	95.83	21.949	96.129	-689.04	16.52	3878.2		E
	95.94	21.949	96.129	-711.895	16.49	3878.2		E
	96.05	21.949	96.129	-720.66	16.45	3878.2		F
	96.16	21.949	96.129	-753.87	16.43	3878.2		F
	96.27	21.949	96.129	-759.64	16.49	3878.2	0.039	G
	477.83	22.064	100.68	-680.069	16.62	3875.6		E
	478.28	22.119	100.645	-706.338	16.55	3875.4		F
	478.73	22.374	100.663	-727.607	16.48	3875.2		F
	479.18	22.429	100.698	-748.876	16.41	3875.1		F
	479.64	22.038	100.68	-760.145	16.50	3874.9		G
209	480.09	22.126	100.715	-700.339	16.40	3876.8		A
	480.54	22.046	100.628	-698.027	16.55	3876.9	0.04	A
	480.99	22.082	100.698	-695.715	16.39	3877.0		A
	481.44	22.117	100.715	-693.403	17.55	3877.0		B
	481.89	22.18	100.663	-691.091	19.44	3877.1		C
	1011.70	21.958	95.900011	-691.635	16.65	3306.7		E
	1012.15	21.975	95.850194	-691.635	16.49	3372.5		E
	1012.60	21.975	95.86034	-715.575	16.42	3329.4		F
	1013.05	22.029	95.854362	-740.45	16.41	3282.3		F
445	1013.50	22.029	95.847834	-755.24	16.34	3349.8	0.035	G

Fig. 8. ESP training.

ranges for glue volume are 0.04 and 0.15 mm<sup>3</sup>, respectively. Any larger (excess of glue) or smaller (Insufficient glue) glue has been defined as a defect. In this experiment, as seen in Appendix A, three insufficient glue types (dispense numbers of 40, 209, and 445) and two excess glue type (dispense numbers of 21 and 210). The data associated with each event represent the qualia of the system state for that specific time. The labels are alphanumerically assigned (e.g., A–Z). For example, in this experiment, defective drops are labeled as C, D, and G. The reason for three different labels is that they have distinctive defect types as far as glue dispensing is concerned (Fig. 8).

3) *Genomes Labeling and Sequential Differentiation*: In the training process, a series of four events of machine state is labeled, the fifth registered event is the glue state (good/defective). Labeling begins with genome length selection. The default genome length is set at five since there are five operational data samples between two consecutive glue dispenses. As the glue state is determined (the fifth event), the previous four machine events are labeled. Each detected machine state is compared with previous labeled events. If the difference is more than 5%, a new event and label is assigned; otherwise, it will be considered as no new state is detected and the same event added to the chain.

For this specific example, the values of “SyringePsi” and HouseVa input parameters determine the state change of the machine. Fig. 8 illustrates the results of running the algorithm for 500 glue dots (ten LCPs), but only five instances of the defect are displayed here. The first trained defect GIP is: “AABCD” for “excess” type of defect’ 0.31 (i.e., defect type D). There is also another GIP for defect type C, (i.e., excess of glue “0.15”), which is “AAABC”. For the “insufficient” type of defect, the GIP is “EEFFG.” Defect type G occurred twice in this example. The results of gene definition, labels, and their sequence of occurrence (DNA) are stored in the process genes pool for the specified process (See Fig. 9).

In conclusion, there is a GIP of EEFFG whose trend is to lower bound defect (insufficient glue volume) and were repeated two times in the training dataset and a GIP of AAABC with a trend to upper bound defect (excess of glue volume). The ESP algorithm will be able to predict the dispensing in real time for both types of defects through sequence

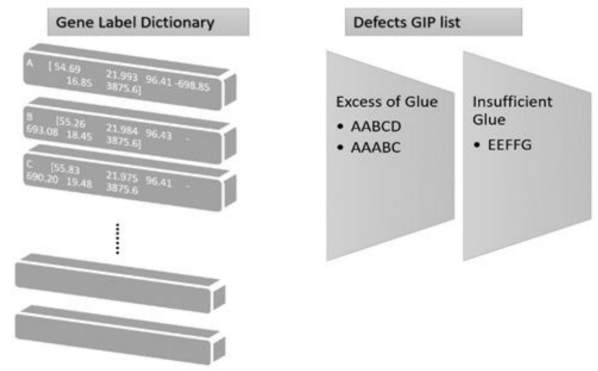


Fig. 9. Database of gene labels dictionary and defects GIP pool.

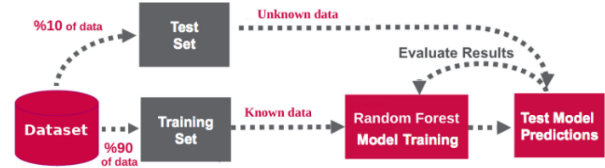


Fig. 10. Model training and predictions.

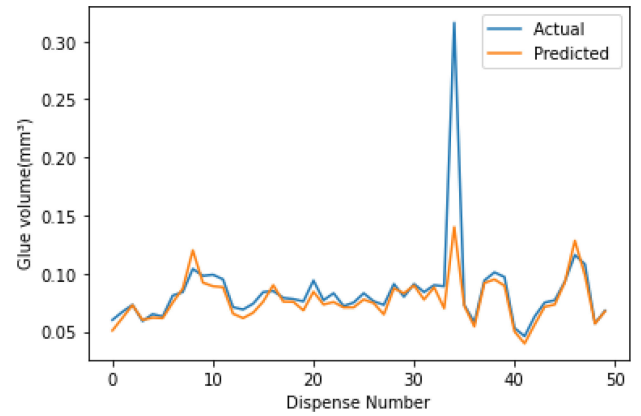


Fig. 11. Predicted data (red) through the RF model and actual data (blue) in the test dataset.

differentiation and cross-check with the database of the Gene labels dictionary and defects GIP list.

## V. ESP PERFORMANCE COMPARED TO OTHER ML ALGORITHMS

The performance of the prediction of ESP is compared with the three most popular data mining and ML [40] methods. Random forest (RF) Regression,  $k$  nearest neighbors (kNN), and multilayer perceptron (MLP) network models were chosen due to their high accuracy and application in similar industrial applications.

RF Regression is one of the most effective and conventional ML models for predictive analytics. The purpose of the algorithm is to ascertain the output from multiple decision trees instead of relying on individual decision trees [41], [42]. Hyperparameters are associated with an RF regression model and used to optimize its performance and include the number of decision trees ( $n_{estimators}$ ) and the number of features considered by each tree when splitting a node ( $max\_feature$ ).



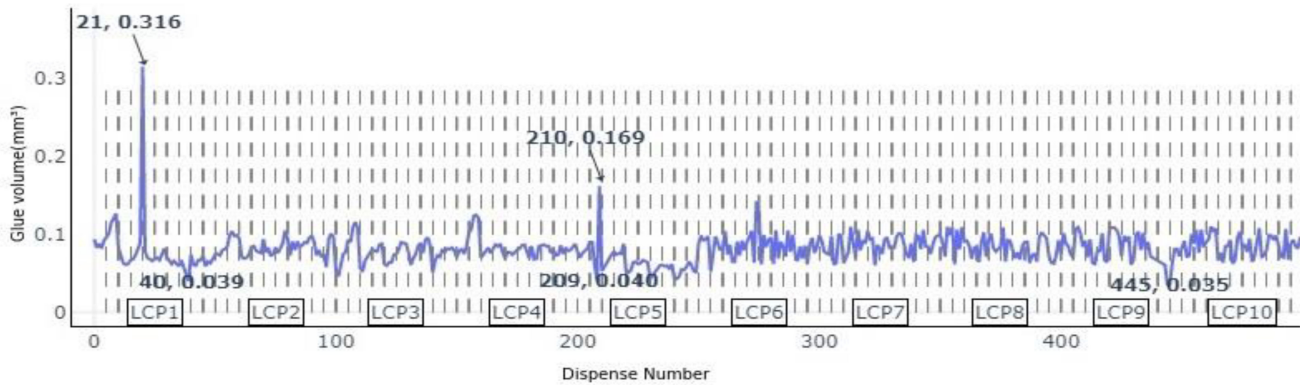


Fig. 12. 500 dispensed glue volumes profile.

TABLE II  
MAE AND MSE BETWEEN THE ACTUAL AND PREDICTED RESULTS  
AFTER TEN TIMES REPETITION

	RF	kNN	MLP
MAE	0.00084	0.00256	0.006541
MSE	7.436861e-06	3.24511e-05	0.00018

TABLE III  
NUMBER OF ERRORS IN PREDICTION

Method	Number of errors
RF	1 (100%)
ESP	0

Hyperparameters optimization is performed by doing an exhaustive search over selected parameter values. In this article, we have set 50 for  $n\_estimator$  and 20 for  $max\_feature$ .

MLP network is one of the most popular and practical architectures of artificial NNs (ANNs). In MLP, every single neuron is connected to its contiguous neuron, with varying impact weights that represent the relativity of the different neuron inputs to the others. The overall weights of the inputs are shifted to the hidden neurons, where it is transformed using an activation function. The other neurons use the outputs of these hidden neurons as inputs, in turn, where they pass through another transformation [43]. The architecture of the MLP that has been employed in this article is based on two hidden layers with eight and ten neurons, respectively. In this model, the weights and biases are modified using the adaptive moment estimation (Adam) optimization function. Moreover, the learning rate was set to 0.01.

To evaluate and compare the performance of the selected methods with the proposed ESP, the same 500 experimental datasets were deployed for the analysis. 90% of the dataset has been selected as training data, assuming at least a defect from each type in the training dataset. And the remaining 10% of the data (i.e., 50 samples) were treated as testing data (unknown or unseen data). Fig. 10 represents the model training and prediction procedure.

Mean absolute error (MAE) and mean squared error (MSE) between the observed and predicted results were used to compare the results. Table II summarizes the outcome of the tests after ten replications. The results show that both types of errors in the RF model are less than kNN and MLP models in this experiment. Then, in the continuation, RF method is selected to be compared with the proposed ESP. However, even with RF regression method, the predictive model will not be able to predict all defects since there are limited observable defects samples in the training dataset (see the RF model results for sample 35 in Fig. 11). One of the pitfalls of classical ML models is their need for large sample sizes (i.e., observable defects)

to build an accurate inferential prediction model. Table III presents the number of errors of both RF and ESP methods in the training dataset.

The comparison between the proposed ESP and RF predictive methods shows that RF method can only reduce uncertainty if the number of observations is statically acceptable. Since the number of observed defects normally in modern industrial processes is very low, such a technique falls short and, with the associated uncertainty, does not provide strong decision support. In contrast, due to the nature of observation and detection of causal relations, the ESP methods do not suffer from uncertainty. The proposed ESP builds a lookup dictionary of genomes based on the occurred defects and corresponding genomes. The proposed ESP method is also superior to similar lookup table solutions used in failure prediction [44] because implementing the EventiC method (dimension reduction) reduces memory usage significantly compared to other lookup table methods. The second advantage of the ESP method is its real-time self-training algorithm. This self-training feature, in comparison to NN offline time-consuming training algorithm, is beneficial.

## VI. CONCLUSION AND FUTURE WORKS

This article described the practical implications of utilizing modern data acquisition techniques, real-time data analytics, and learning methods in industrial environments. The challenges posed by the complexity and timing demand of the case study motivated us to go beyond the existing ML and AI techniques that could not offer satisfactory results on their own. The new technique for the real-time sequencing strategy not only has the capability to predict events for key performance indicators of a given industrial system but also provides an alternative state-space description of industrial processes in the identification of root causes of suboptimal performances. A comparison of the proposed ESP method and the RF machine learning method was conducted in their performance and level of confidence. The results have helped

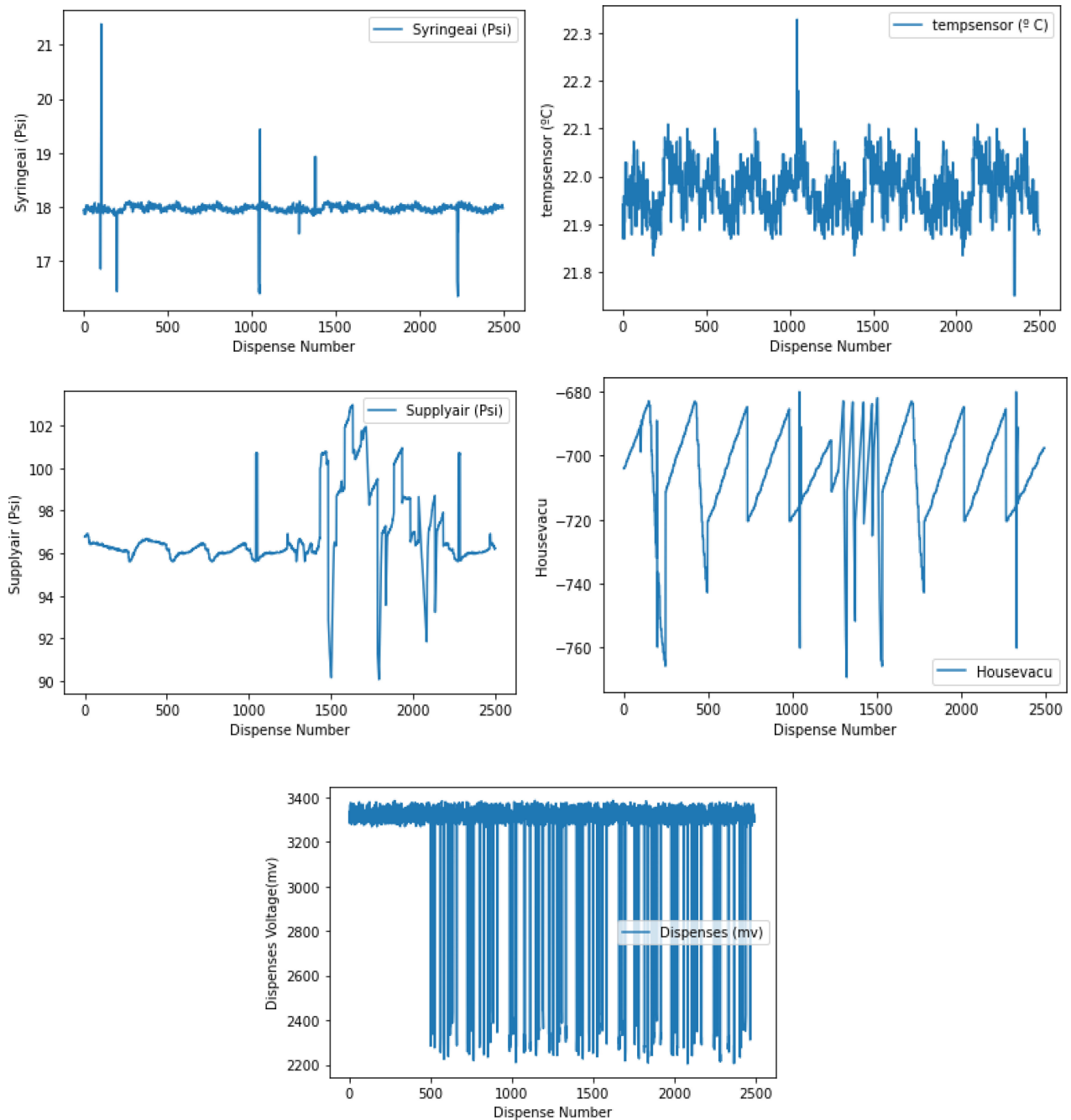


Fig. 13. Dispense machine operational and environmental sensor parameters.

increase confidence that by deploying ML and AI in manufacturing/industrial processes, there is a possibility to build zero defects–zero waste processes.

The proposed theory of process GIP takes its inspiration from biology and genetic science, thus creating a process for labeling and establishing sequential event differentiation from observed information. These labeled genomes of the process are being used to predict the events according to their occurrence sequence. GIP can be classified as a Granular computing (GrC) technique. GrC is an approach for reasoning a system and dividing information into smaller pieces (Genes) to see if they differ on a granular (trigger threshold) level. This approach could become a new tool for deep and reinforcement

learning. A future pathway for research is to explore its ability to rationalize the existing imbalanced data labeling during training datasets by reducing reliance on statistical means. This problem is prevalent in manufacturing environments where potentially a mixture of weighted factors could cause serious misinterpretation and loss of productivity.

One of the applications of GIP to be explored is in the enhancement of learning and reasoning capabilities of cognitive digital twins (CDTs). The gene pool created by GIP real-time gene recognition throughout the learning process allows the nature of the physical process to be transformed into reusable and referenceable knowledge. The gene sequencing (DNA of the process) can be a powerful tool for interpreting

and demonstrating the knowledge graphs in CDT. Another potential area to explore is the role of GIP in simplifying and relaxing rule-based models (e.g., [45] and [46]).

#### APPENDIX A

##### 500 DISPENSED GLUE VOLUMES PROFILE

See Fig. 12.

#### APPENDIX B

##### DISPENSE MACHINE OPERATIONAL AND ENVIRONMENTAL SENSOR PARAMETERS

See Fig. 13.

#### REFERENCES

- [1] R. Sun and C.-L. Giles, (Eds.), *Sequence Learning: Paradigms, Algorithms, and Applications*. Berlin, Germany: Springer, 2003.
- [2] N. G. Nguyen *et al.*, "DNA sequence classification by convolutional neural network," *J. Biomed. Sci. Eng.*, vol. 9, no. 5, pp. 280–286, 2016.
- [3] C. Zhou, B. Cule, and B. Goethals, "Pattern based sequence classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1285–1298, May 2016.
- [4] "Pandakekok, Gene Prediction," Dec. 2016. [Online]. Available: [https://commons.wikimedia.org/wiki/File:Gene\\_Prediction.png#filehistory](https://commons.wikimedia.org/wiki/File:Gene_Prediction.png#filehistory)
- [5] J. Yin, N. Wang, and A. N. Perakis, "A real-time sequential ship roll prediction scheme based on adaptive sliding data window," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 12, pp. 2115–2125, Dec. 2018.
- [6] C. C. Gonzalez and M. R. Burke, "Motor sequence learning in the brain: The long and short of it," *Neuroscience*, vol. 389, pp. 85–98, Oct. 2018.
- [7] J. Crivelli-Decker, L.-T. Hsieh, A. Clarke, and C. Ranganath, "Theta oscillations promote temporal sequence learning," *Neurobiol. Learn. Memory*, vol. 153, pp. 92–103, Sep. 2018.
- [8] R. Baker, M. Dexter, T. E. Hardwicke, A. Goldstone, and Z. Kourtzi, "Learning to predict: Exposure to temporal sequences facilitates prediction of future events," *Vis. Res.*, vol. 99, pp. 124–133, Jun. 2014.
- [9] K. Hansson, S. Yella, M. Dougherty, and H. Fleyeh, "Machine learning algorithms in heavy process manufacturing," *Amer. J. Intell. Syst.*, vol. 6, no. 1, pp. 1–13, 2016.
- [10] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.
- [11] S. Das, A. Dey, A. Pal, and N. Roy, "Applications of artificial intelligence in machine learning: Review and prospect," *Int. J. Comput. Appl.*, vol. 115, no. 9, pp. 31–41, 2015.
- [12] Y. Liu, J.-W. Bi, and Z.-P. Fan, "Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms," *Expert Syst. Appl.*, vol. 80, pp. 323–339, Sep. 2017. [Online]. Available: [www.scopus.com](http://www.scopus.com)
- [13] A. Dey, "Machine learning algorithms: A review," *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 3, pp. 1174–1179, 2016.
- [14] G. Khodabandelou, C. Hug, R. Deneckère, and C. Salinesi, "Supervised vs. unsupervised learning for intentional process model discovery," in *Enterprise, Business-Process and Information Systems Modeling*. Berlin, Germany: Springer, 2014, pp. 215–229.
- [15] F. Wörgötter and B. Porr, "Temporal sequence learning, prediction, and control: A review of different models and their relation to biological mechanisms," *Neural Comput.*, vol. 17, no. 2, pp. 245–319, 2005.
- [16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
- [17] K. George and P. Mutalik, "A multiple model approach to time-series prediction using an online sequential learning algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 5, pp. 976–990, May 2019.
- [18] S. Cho and H. Foroosh, "A temporal sequence learning for action recognition and prediction," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2018, pp. 352–361.
- [19] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance-based time series classification," *Data Mining Knowl. Discov.*, vol. 33, no. 2, pp. 378–412, 2019.
- [20] H. Wang, Q. Zhang, J. Wu, S. Pan, and Y. Chen, "Time series feature learning with labeled and unlabeled data," *Pattern Recognit.*, vol. 89, pp. 55–66, May 2019.
- [21] J. P. Nóbrega and A. L. Oliveira, "A sequential learning method with Kalman filter and extreme learning machine for regression and time series forecasting," *Neurocomputing*, vol. 337, pp. 235–250, Apr. 2019.
- [22] X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari, and M. Cheriet, "KRNN: k Rare-class nearest neighbour classification," *Pattern Recognit.*, vol. 62, pp. 33–44, Feb. 2017.
- [23] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2018.
- [24] A. de Haro-García, J. Pérez-Rodríguez, and N. García-Pedrajas, "Combining three strategies for evolutionary instance selection for instance-based learning," *Swarm Evol. Comput.*, vol. 42, pp. 160–172, Oct. 2018.
- [25] B. Krawczyk, I. Triguero, S. García, M. Woźniak, and F. Herrera, "Instance reduction for one-class classification," *Knowl. Inf. Syst.*, vol. 59, no. 3, pp. 601–628, 2019.
- [26] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, and D. Tao, "A regularization approach for instance-based superset label learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 967–978, Mar. 2018.
- [27] Y. Chen and Y. Hao, "A feature weighted support vector machine and k-nearest neighbor algorithm for stock market indices prediction," *Expert Syst. Appl.*, vol. 80, pp. 340–355, Sep. 2017.
- [28] H. Ren, Z. Ye, and Z. Li, "Anomaly detection based on a dynamic Markov model," *Inf. Sci.*, vol. 411, pp. 52–65, Oct. 2017.
- [29] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 823–839, May 2012.
- [30] M. Danishvar, A. Mousavi, and P. Broomhead, "EventC: A real-time unbiased event-based learning technique for complex systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 55, no. 5, pp. 1649–1662, May 2020.
- [31] S. Tavakoli, A. Mousavi, and P. Broomhead, "Event tracking for real-time unaware sensitivity analysis (EventTracker)," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 2, pp. 348–359, Feb. 2013.
- [32] N. I. Mehrkish, "The mathematical model of glue dispersion and amelioration its performance in presence of optimal controllers," Ph.D. Dissertation, Dept. Electron. Comput. Eng., Brunel Univ., London, U.K., 2017.
- [33] X. B. Chen, G. Schoenau, and W. J. Zhang, "Dynamic modeling of the flow rate in time-pressure fluid dispensing processes," in *Proc. IPACK*, vol. 1, 2001, pp. 516–520.
- [34] M. Abd Al Rahman and A. Mousavi, "A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry," *IEEE Access*, vol. 8, pp. 183192–183271, 2020.
- [35] T. Vafeiadis, N. Dimitriou, D. Ioannidis, T. Wotherspoon, G. Tinker, and D. Tzovaras, "A framework for inspection of dies attachment on PCB utilizing machine learning techniques," *J. Manag. Anal.*, vol. 5, no. 2, pp. 81–94, Feb. 2018.
- [36] S. Fonseca and J. A. K. S. Jayasinghe, "Feature extraction and template matching algorithms classification for PCB fiducial verification," *J. Achievements Mater. Manuf. Eng.*, vol. 86, no. 1, pp. 14–32, 2018.
- [37] N. Dimitriou *et al.*, "Fault diagnosis in microelectronics attachment via deep learning analysis of 3-D laser scans," *IEEE Trans. Ind. Electron.*, vol. 67, no. 7, pp. 5748–5757, Jul. 2020, doi: [10.1109/TIE.2019.2931220](https://doi.org/10.1109/TIE.2019.2931220).
- [38] D. Maturana and S. A. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 922–928.
- [39] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.
- [40] N. Settouti, M. E. A. Bechar, and M. A. Chikh, "Statistical comparisons of the top 10 algorithms in data mining for classification task," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 1, pp. 46–51, 2016.
- [41] T. Ahmad and H. Chen, "Nonlinear autoregressive and random forest approaches to forecasting electricity load for utility energy management systems," *Sustain. Cities Soc.*, vol. 45, pp. 460–473, Feb. 2019.
- [42] Y. Cui, S. Ahmad, and J. Hawkins, "Continuous Online sequence learning with an unsupervised neural network model," *Neural Comput.*, vol. 28, no. 11, pp. 2474–2504, 2016.
- [43] E. Heidari, M. A. Sobati, and S. Movahedirad, "Accurate prediction of nanofluid viscosity using a multilayer perceptron artificial neural network (MLP-ANN)," *Chemometr. Intell. Lab. Syst.*, vol. 155, pp. 73–85, Jul. 2016.
- [44] Y. S. Kim, "Failure prediction of metal oxide varistor using nonlinear surge look-up table based on experimental data," *Trans. Elect. Electron. Mater.*, vol. 16, no. 6, pp. 317–322, 2015.

- [45] Z. Huang, M. Li, A. Mousavi, M. Danishvar, and Z. Wang, "EGEP: An event tracker enhanced gene expression programming for data driven system engineering problems," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 117–126, Apr. 2019.
- [46] M. Razgon and A. Mousavi, "Relaxed rule-based learning for automated predictive maintenance: Proof of concept," *Algorithms*, vol. 13, no. 9, p. 219, 2020.



**Alireza Mousavi** (Senior Member, IEEE) received the B.Sc. degree in industrial engineering and the Ph.D. degree in system engineering from Brunel University London, Uxbridge, U.K.

He is a Reader with the College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge, U.K. His research interests are in smart supervisory control and data acquisition systems applied to real-time systems modeling and optimization. The key areas of application are stochastic modeling, ontology alignment, and sensor networks.

Dr. Mousavi is a Senior Member of IET.



**Morad Danishvar** received the Ph.D. degree in system engineering from Brunel University London, Uxbridge, U.K., in 2015.

He is a Research Scientist with the System Engineering Research Group, Brunel University London, where he is also the Deputy Director of Digital Manufacturing Centre. His research interests are in data science, data analytics and machine learning, AI, digital manufacturing, DSS, real-time systems modeling, and optimization.



**Sebelan Danishvar** received the Ph.D. degree in electrical and computer engineering from Tarbiat Modares University, Tehran, Iran, in 2007.

He currently collaborates and works on EU Horizon 2020 Z-Factor and HISTEED projects. His current research activities are modeling and simulation, machine learning, and image processing.