



(51) International Patent Classification:

G06N 3/12 (2006.01) G06N 5/00 (2006.01)
G06N 20/20 (2019.01) G06N 7/00 (2006.01)
G06N 3/00 (2006.01) G06N 20/10 (2019.01)
G06N 3/04 (2006.01) G06N 3/08 (2006.01)

(21) International Application Number:

PCT/EP2020/053138

(22) International Filing Date:

07 February 2020 (07.02.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/805,721 14 February 2019 (14.02.2019) US

(71) Applicants: **BAYER CROSCIENCE LP** [US/US]; 800 North Lindbergh Blvd, St. Louis, MO 63167 (US). **BAYER AKTIENGESELLSCHAFT** [DE/DE]; Kaiser-Wilhelm-Allee 1, 51373 Leverkusen (DE).

(72) Inventors: **ALLY, Dilara**; 5864 13th St, Sacramento, CA 95822 (US). **ARMSTRONG, Don**; 8417 Grinnell Way, Sacramento, CA 95826 (US). **GRAVES, Christopher**; 1340 Taylor St, San Francisco, CA 941008 (US). **HEBING, Lukas**; Merheimer Str. 94, 50733 Köln (DE). **NIEDENFÜHR, Sebastian**; Peter-Kütter Weg 11, 51061 Köln (DE). **TAYLOR, Colleen**; 112 Kennar Way, Folsom, CA 95630 (US). **NEWMAN, Adam**; 7940 Pocket Rd. #7, Sacramento, CA 95831 (US).

(74) Agent: **BIP PATENTS**; Alfred-Nobel-Str. 10, 40789 Monheim am Rhein NRW (DE).

(54) Title: MACHINE LEARNING BASED ALGORITHMIC PROCEDURE

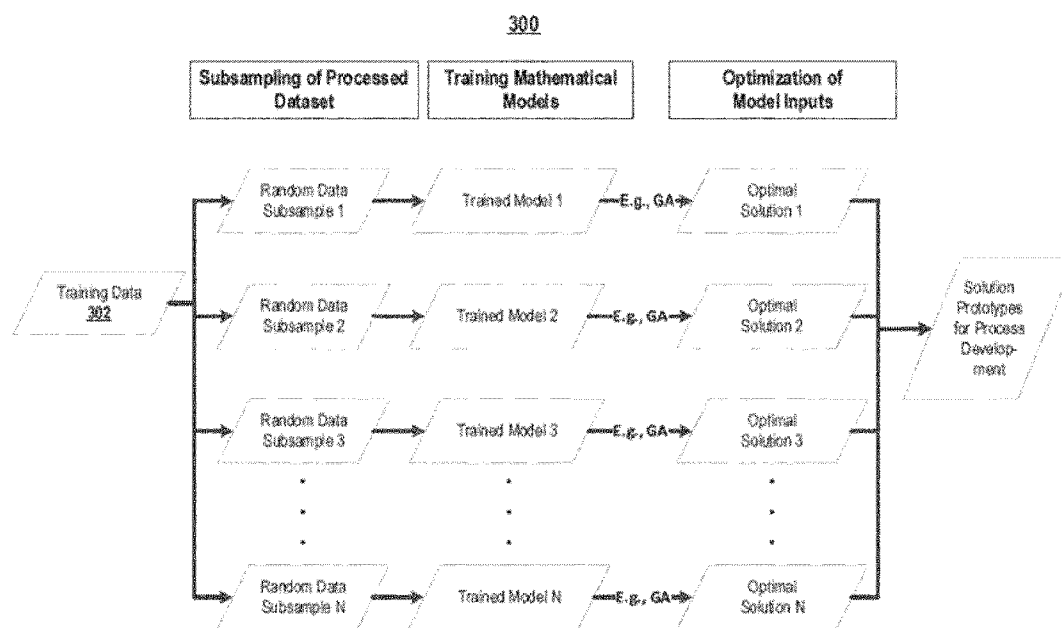


FIG. 3

(57) **Abstract:** The present disclosure generally relates to systems and methods for optimizing a fermentation process. An exemplary method for optimizing a fermentation process comprises: subsampling a training dataset to obtain a plurality of subsampled datasets, wherein the training dataset includes data related to one or more previous instances of the fermentation process; training a plurality of mathematical models based on the plurality of subsampled datasets; identifying a plurality of sets of fermentation conditions based on the plurality of trained mathematical models; identifying one or more clusters of sets of fermentation conditions based on the plurality of sets of fermentation conditions; obtaining one or more fermentation design prototypes based on the one or more clusters; and determining a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.



(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

MACHINE LEARNING BASED ALGORITHMIC PROCEDURE

FIELD OF THE INVENTION

[0001] The present disclosure relates generally to optimizing fermentation
5 processes, and more specifically to optimizing fermentation processes based on ensemble machine learning algorithms.

BACKGROUND

[0002] Traditionally, fermentation processes are studied by conducting
10 fermentation experiments under a controlled environment. This approach has several drawbacks. For example, the approach relies only on limited data gathered from controlled experiments and does not take advantage of rich data from previous fermentation processes that occurred outside the laboratories. Further, because the experiments are designed with limited insight into the actual processes, it is difficult to ensure that the gathered data can
15 reveal all significant statistical associations among the relevant factors. Further, when new data is gathered, there is no systematic way to revise existing inferences based on the new data.

[0003] In comparison, leveraging modeling and optimization methods to uncover optimal process conditions can be advantageous. Specifically, machine learning models can
20 make use of all relevant and accessible historical data and have the flexibility to “learn” any relationships among variables. For example, a neural network can be trained based on a large amount of data gathered from historical fermentation processes and can be retrained using new data.

[0004] However, applying modeling and optimization methods to study
25 fermentation still presents several challenges. First, it can be difficult to generate a diverse set of solutions (e.g., optimal fermentation conditions) from an optimization method, which usually yields only one optimal solution. A diverse set of possible optimal solutions, however, are generally critical to guide downstream analysis and guide future empirical data collection to study unexplored aspects of the fermentation process. Further, especially predictions from
30 a machine learning model often lack statistical confidence measures that can help inform decisions related to the optimization and control of the process. Accordingly, it is still a

challenge to leverage machine learning models and optimization methods to conduct a comprehensive and guided optimization of fermentation processes.

BRIEF SUMMARY

[0005] The invention is directed to systems, devices, and methods for training an ensemble (or collection) of models, preferable machine learning models (e.g., a collection of neural networks, a collection of support vector machines), rather than a single machine learning model, based on historical fermentation data to obtain statistical descriptions of a fermentation process. Subsequent mathematical optimization of these trained models results in an ensemble of sets of optimal fermentation conditions. The ensemble of optimal fermentation conditions can lead to a diverse set of process design prototypes and process control strategies.

[0006] The invention is further directed to systems, devices, and methods for applying analytical methods such as clustering and multivariate data analysis to the ensemble of sets of optimal fermentation conditions to gain new insights into the fermentation process. For example, the analysis can uncover key variables of the fermentation process for further experimentation and exploration. As another example, the analysis can identify statistical confidence measures and prediction intervals of the design prototypes, which can be used to guide experimentation design and further optimization of the process.

[0007] In some embodiments, a computer-enabled method for optimizing a fermentation process comprises: subsampling a training dataset to obtain a plurality of subsampled datasets, wherein the training dataset includes data related to one or more previous instances of the fermentation process; training a plurality of mathematical models based on the plurality of subsampled datasets; identifying a plurality of sets of fermentation conditions based on the plurality of trained mathematical models; identifying one or more clusters of sets of fermentation conditions based on the plurality of sets of fermentation conditions; obtaining one or more fermentation design prototypes based on the one or more clusters; and determining a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

[0008] In some embodiments, the method further comprises performing a new instance of the fermentation process based on the candidate fermentation condition.

[0009] In some embodiments, the method further comprises obtaining a new training dataset based on the new instance of the fermentation process; and training the plurality of mathematical models based on the new training dataset.

[0010] In some embodiments, the training data comprises data related to one or more fermentation conditions of the fermentation process.

[0011] In some embodiments, the one or more fermentation conditions comprise one or more media ingredients, one or more physical conditions, online data, one or more process data points, or any combination thereof.

[0012] In some embodiments, the one or more media ingredients comprise: water content, carbon content, nitrogen content, nitrogen sources (e.g., ammonia, urea, protein, amino acids), carbohydrates (simple (e.g., glucose, sucrose, fructose, oligosaccharides) and complex (e.g., maltodextrins, starch)), trace elements (e.g., magnesium, potassium, iron, zinc, manganese, calcium, copper, potassium, phosphorus, sodium, potassium, sulfate, chloride), vitamins (biotin, folic acid, thiamine), complex media (commercially available, soy/corn flours or meals, yeast extracts, molasses, corn steep liquor, cottonseed-based flour), small organic acids (lactate, acetate), amino acids, precursors for a desired product (e.g., glutamate, pyruvate), proteins, peptides, fat or fatty acids, buffer components (e.g., CaCO_3 , phosphate-buffers) and anti-foam, or any combination thereof.

[0013] In some embodiments, the one or more physical conditions comprise: one or more physical conditions comprise: pH, temperature, sparge rate, CO_2 flow rate, O_2 flow rate, sterilization time, sterilization temperature, addition of components after sterilization, or any combination thereof.

[0014] In some embodiments, the one or more online data comprise: measurements of pH, temperature, sparge rate, dissolved oxygen, CO_2 flow rate, O_2 flow rate, respiration-carbon evolution rate (CER), oxygen uptake rate (OUR), mixing rate, ingredient feed rates, ingredient feed times, temperature profiles, antifoam, sterilization time, sterilization temperature, total process time or any combination thereof.

[0015] In some embodiments, the one or more process data points comprise features, which are extracted from time-course data or online probe measurements (like the maximum- or minimum value, the slope, duration of specific features like the duration until a certain value is reached, the duration in which a value stays within a certain range, a calculated lag phase (e.g., from oxygen sensor measurements) or combinations thereof).

[0016] In some embodiments, the training data comprises data related to one or more fermentation outcomes of the fermentation process.

[0017] In some embodiments, one or more fermentation outcomes comprise: one or more formed products, one or more measures for characterizing biomass, one or more
5 properties of the solution, one or more biological assay performance, one or more attributes of the product, one or more concentration or amount of by-products or impurities, one or more costs of ingredients, or any combination thereof.

[0018] In some embodiments, the one fermentation outcomes comprise: absolute amount or relative measure of the formed product (e.g., biological chemistry, biomolecules,
10 proteins with a function of interest, whole cells, metabolic products (metabolites), biotransformation products), additional measures characterizing biomass (cell dry weight, spores count, colony forming units (CFU), packed cell volume (PCV), degree of sporulation), biological assay performance (e.g., field trial, *in vitro* assay, etc), properties of the solution (viscosity), attributes of the product or product mixtures (spore fitness, post-translational
15 modifications, protein folding, aggregation state of protein, chiral properties, alternative spliced versions, product mixture fractions, measures for shelf-life), concentration or amount of by-products or impurities (host-cell protein, residual media components, precipitated compounds), costs (for materials, media ingredients and operation) or a combination thereof.

[0019] In some embodiments, the training data is derived from one or more
20 previous fermentation experiments.

[0020] In some embodiments, subsampling the training dataset comprises subsampling the training dataset via a bootstrap resampling technique.

[0021] In some embodiments, the plurality of mathematical models are machine learning models.

25 [0022] In some embodiments, the plurality of machine learning models comprises a plurality of neural networks.

[0023] In some embodiments, the plurality of machine learning models comprises a plurality of support vector machines.

[0024] In some embodiments, the plurality of mathematical models comprises:
30 one or more Gaussian process models, one or more radial basis function networks, one or more partial-least-squares, generalized linear models, one or more support vector machines,

one or more nearest neighbor algorithms, one or more random forest algorithms, one or more neural networks, or any combination thereof.

[0025] In some embodiments, the plurality of mathematical models comprises a first mathematical model and a second mathematical model; the plurality of subsampled
5 datasets comprises a first subsampled dataset and a second subsampled dataset; and training the plurality of mathematical models based on the plurality of subsampled datasets comprises: training the first mathematical model based the first subsampled dataset and training the second mathematical model based on the second subsampled dataset.

[0026] In some embodiments, each of the plurality of trained mathematical models
10 is configured to provide a predicted fermentation outcome based on a set of fermentation conditions.

[0027] In some embodiments, identifying the plurality of sets of fermentation conditions based on the plurality of trained mathematical models comprises: for each trained mathematical model of the plurality of trained mathematical models, identifying a set of
15 fermentation conditions based on a optimization algorithm which maximizes or minimizes an objective function.

[0028] In some embodiments, the optimization algorithm is a population-based search heuristics.

[0029] In some embodiments, the optimization algorithm is a genetic algorithm.

[0030] In some embodiments, the optimization algorithm is configured to identify
20 a set of fermentation conditions that optimizes the objective function.

[0031] In some embodiments, the objective function is calculated based on one or more fermentation outcomes predicted by a trained mathematical model, a cost value, or any combination thereof.

[0032] In some embodiments, an objective function comprises a user-specified
25 formula.

[0033] In some embodiments, identifying the one or more clusters comprises: performing an unsupervised clustering algorithm on the plurality of sets of fermentation conditions.

[0034] In some embodiments, identifying the one or more clusters comprises:
30 performing k-means or any other unsupervised clustering algorithm (e.g., x-means,

hierarchical clustering, mixture of gaussians) on the plurality of sets of fermentation conditions.

[0035] In some embodiments, each fermentation design prototype of the one or more fermentation design prototypes corresponds to a centroid of the corresponding cluster of the one or more clusters.

[0036] In some embodiments, the method further comprises calculating a prediction intervals and a number of members associated with each fermentation design prototype based on the corresponding cluster.

[0037] In some embodiments, determining a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes comprises: selecting the candidate fermentation condition based on the prediction interval of each fermentation design prototype and the number of members of the associated cluster.

[0038] In some embodiments, the method further comprises: performing a multivariate analysis based on the plurality of sets of fermentation conditions to understand correlations and important or not important key variables in the identified fermentation conditions.

[0039] In some embodiments, an exemplary system comprises: one or more processors; a memory; and one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for: subsampling a training dataset to obtain a plurality of subsampled datasets, wherein the training dataset includes data related to one or more previous instances of a fermentation process; training a plurality of mathematical models based on the plurality of subsampled datasets; identifying a plurality of sets of fermentation conditions based on the plurality of trained mathematical models; identifying one or more clusters of sets of fermentation conditions based on the plurality of sets of fermentation conditions; obtaining one or more fermentation design prototypes based on the one or more clusters; and determining a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

[0040] In some embodiments, an exemplary non-transitory computer-readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by one or more processors of an electronic device, cause the electronic device to: subsample a training dataset to obtain a plurality of subsampled

datasets, wherein the training dataset includes data related to one or more previous instances of the fermentation process; train a plurality of mathematical models based on the plurality of subsampled datasets; identify a plurality of sets of fermentation conditions based on the plurality of trained mathematical models; identify one or more clusters of sets of fermentation
5 conditions based on the plurality of sets of fermentation conditions; obtain one or more fermentation design prototypes based on the one or more clusters; and determine a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

10

DESCRIPTION OF THE FIGURES

[0041] **FIG. 1** illustrates an exemplary process for optimizing a fermentation process, according to various embodiments.

[0042] **FIG. 2** illustrates an exemplary process for optimizing a fermentation process, according to various embodiments.

15

[0043] **FIG. 3** illustrates a number of exemplary steps and the corresponding data in a process for obtaining optimal fermentation conditions and solution prototypes, according to various embodiments.

[0044] **FIG. 4A** illustrates an exemplary trained neural network, according to various embodiments.

20

[0045] **FIG. 4B** illustrates an exemplary neural network/genetic algorithm model for obtain an optimal solution to a neural network, according to various embodiments.

[0046] **FIG. 5** illustrates an exemplary process for analyzing optimal fermentation conditions, according to various embodiments.

25

[0047] **FIG. 6A** illustrates plotting an exemplary ensemble of optimal solutions into principal components space, according to various embodiments.

[0048] **FIG. 6B** illustrates plotting an exemplary ensemble of optimal solutions into principal components space, according to various embodiments.

[0049] **FIG. 7A** illustrates plotted PCA loadings of an exemplary ensemble of optimal solutions, according to various embodiments.

30

[0050] **FIG. 7B** illustrates plotted PCA loadings of an exemplary ensemble of optimal solutions, according to various embodiments.

[0051] **FIG. 8** illustrates an elbow curve showing the score of the k-means clustering of an exemplary ensemble of optimal solutions, according to various embodiments.

[0052] **FIG. 9** illustrates an exemplary 95% prediction interval of each solution prototype and extent of convergence among model ensemble, according to various
5 embodiments.

[0053] **FIG. 10** illustrates an exemplary process for optimizing a fermentation process, according to various embodiments.

[0054] **FIG. 11** depicts an exemplary electronic device according to various
10 embodiments.

DETAILED DESCRIPTION

[0055] The invention is directed to systems, devices, and methods for training an ensemble (or collection) of models such as machine learning models (e.g., a collection of neural networks, a collection of support vector machines), rather than a single machine
15 learning model, based on historical fermentation data to obtain statistical descriptions of a fermentation process. Subsequent mathematical optimization of these trained models results in an ensemble of sets of optimal fermentation conditions. The ensemble of optimal fermentation conditions can lead to a diverse set of process design prototypes and process control strategies.

[0056] The invention is further directed to systems, devices, and methods for
20 applying analytical methods such as clustering and multivariate data analysis to the ensemble of sets of optimal fermentation conditions to gain new insights into the fermentation process. For example, the analysis can uncover key variables of the fermentation process for further experimentation and exploration. As another example, the analysis can identify statistical
25 confidence measures and prediction intervals of the design prototypes, which can be used to guide experimentation design and further optimization of the process.

[0057] The following description is presented to enable a person of ordinary skill
in the art to make and use the various embodiments. Descriptions of specific devices,
techniques, and applications are provided only as examples. Various modifications to the
30 examples described herein will be readily apparent to those of ordinary skill in the art, and the general principles defined herein may be applied to other examples and applications without departing from the spirit and scope of the various embodiments. Thus, the various

embodiments are not intended to be limited to the examples described herein and shown, but are to be accorded the scope consistent with the claims.

[0058] Although the following description uses terms “first,” “second,” etc. to describe various elements, these elements should not be limited by the terms. These terms are only used to distinguish one element from another. For example, a first model could be termed a second model, and, similarly, a second model could be termed a first model, without departing from the scope of the various described embodiments. The first model and the second model are both graphical representations, but they are not the same graphical representation.

[0059] The terminology used in the description of the various described embodiments herein is for the purpose of describing particular embodiments only and is not intended to be limiting. As used in the description of the various described embodiments and the appended claims, the singular forms “a,” “an,” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms “includes,” “including,” “comprises,” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0060] The term “if” is, optionally, construed to mean “when” or “upon” or “in response to determining” or “in response to detecting,” depending on the context. Similarly, the phrase “if it is determined” or “if [a stated condition or event] is detected” is, optionally, construed to mean “upon determining” or “in response to determining” or “upon detecting [the stated condition or event]” or “in response to detecting [the stated condition or event],” depending on the context.

[0061] **FIG. 1** illustrates an exemplary process 100 for optimizing a fermentation process, according to various embodiments. The process 100 starts by obtaining raw data related to previous fermentation processes and preprocessing the raw data to generate a training dataset. The training data can include any data related to the fermentation process to be optimized, such as fermentation conditions and fermentation outcomes, as discussed further below.

[0062] As shown in **FIG. 1**, the training data is provided to an ensemble algorithm 102. The ensemble algorithm comprises an ensemble of statistical learning models and an ensemble of optimization algorithms. An ensemble algorithm is a “meta-algorithm” that improves performance by combining predictions from a large set of machine learning models and optimization algorithms. For example, the ensemble algorithm can include N number of neural networks and one or more optimization algorithms (or differently tuned optimizer’s parameters). It should be appreciated, however, that any type of models and any type of optimization algorithms can be used as part of the ensemble algorithm.

[0063] The training dataset is subsampled into an ensemble of datasets. The ensemble of datasets is used to train the ensemble of models (e.g., neural networks) in the ensemble algorithm 102. Subsequently, the ensemble of trained machine learning models is optimized using the ensemble of optimization algorithms. The result of the ensemble algorithm is an ensemble of optimal solutions. Each optimal solution can comprise a set of optimal fermentation conditions (e.g., an optimal temperature, an optimal amount of media, a media composition).

[0064] As shown in **FIG. 1**, clustering (or some other type of unsupervised learning technique) can be used to aggregate similar optimal solutions. Clustering of solutions generates a set of distinct design prototypes for experimentation and validation. Cluster size can indicate statistical confidence in the prediction via bootstrapping, as discussed below. Additional analysis can be performed to gain insights into the fermentation process. For example, multivariate statistical analysis can be performed to identify key process variables for further experimentation and exploration. As another example, prediction intervals for the design prototypes can be identified. Empirical validation can produce additional process data, which can be used to retrain the ensemble algorithm 102.

[0065] **FIG. 2** illustrates an exemplary process 200 for optimizing a fermentation process, according to various embodiments. Process 200 is performed, for example, using one or more electronic devices implementing a software platform. In some examples, process 200 is performed using a client-server system, and the blocks of process 200 are divided up in any manner between the server and client device(s). Thus, while portions of process 200 are described herein as being performed by a particular electronic device, it will be appreciated that process 200 is not so limited. In process 200, some blocks are, optionally, combined, the order of some blocks is, optionally, changed, and some blocks are, optionally, omitted. In

some examples, additional steps may be performed in combination with the process 200. Accordingly, the operations as illustrated (and described in greater detail below) are exemplary by nature and, as such, should not be viewed as limiting.

[0066] At block 202, an exemplary system (e.g., comprising one or more electronic devices) obtains training data 206. Training data generally refers to a set of data relevant to the quantity or quantities to be predicted by a machine learning model, where the set of data has been directly observed or are otherwise known or approximated. The training data 206 can include any data related to the fermentation process to be optimized (e.g., a fermentation process involving *paenibacillus*). For example, the training data 206 can include fermentation conditions (e.g., media ingredients, physical conditions, or any combinations thereof) and fermentation outcomes (e.g., amount formed product, attributes of the product, concentration or amount by by-products or impurities, or any combinations thereof).

[0067] In some embodiments, the data related to fermentation conditions comprise data related to media ingredients (water content, carbon content, nitrogen content, nitrogen sources (e.g., ammonia, urea, protein, amino acids), carbohydrates (simple (e.g., glucose, sucrose, fructose, oligosaccharides) and complex (e.g., maltodextrins, starch)), trace elements (e.g., magnesium, potassium, iron, zinc, manganese, calcium, copper, potassium, phosphorus, sodium, potassium, sulfate, chloride), vitamins (biotin, folic acid, thiamine), complex media (commercially available, soy/corn flours or meals, yeast extracts, molasses, corn steep liquor, cottonseed-based flour), small organic acids (lactate, acetate), amino acids, precursors for a desired product (e.g., glutamate, pyruvate), proteins, peptides, fat or fatty acids, buffer components (e.g., CaCO_3 , phosphate-buffers) and anti-foam, or any combination thereof.

[0068] In some embodiments, the the data related to online data (pH, temperature, sparge rate, dissolved oxygen, CO_2 flow rate, O_2 flow rate, respiration-carbon evolution rate (CER), oxygen uptake rate (OUR), mixing rate, ingredient feed rates, ingredient feed times, temperature profiles, addition of certain media components before or after sterilization, antifoam, sterilization time, sterilization temperature, total process time, or any combination thereof.

[0069] In some embodiments, the the data related physical conditions comprise: pH, temperature, sparge rate, CO_2 flow rate, O_2 flow rate, sterilization time, sterilization temperature, addition of components after sterilization or any combination thereof. In some embodiments, the fermentation output comprises data related to: amount of formed product

(e.g., biological chemistry, biomolecules, proteins with a function of interest, whole cells, metabolic products (metabolites), biotransformation products), spores, colony forming units (CFU), packed cell volume (PCV), degree of sporulation, viscosity, measures for shelf-life, or biological assay performance (e.g., field trial, in vitro assay, etc), viscosity, attributes of the product or product mixtures (spore fitness, post-translational modifications, protein folding, aggregation state of protein, chiral properties, alternative spliced versions, product mixture fractions), concentration or amount of by-products or impurities (host-cell protein, residual media components, precipitated compounds), or any combination thereof.

[0070] The training data 206 can be compiled from a plurality of data sources, such as published databases, historical and proprietary data sources from the same process or similar processes, and/or preliminary empirical data. Further, the training data 206 can further be collected from different environments, such as micro-reactor, bioreactor, pilot plants, or any combination thereof.

[0071] In some embodiments, block 202 includes data preprocessing steps such as quality control, transformation, imputation, simulation, and removal to generate the training data 206. For example, because the raw fermentation data may be of multiple types and from multiple sources, the raw data may need to be integrated and preprocessed (e.g., such that they have consistent formats, structures, and linkages) to create the training data set 206. As another example, raw fermentation data may need to be converted to a format compatible with the machine learning model to be trained. For example, the system may normalize the raw data by creating scalers (e.g., scale each variable to have a means of zero and variance of one).

[0072] At block 204, the system selects an untrained model ensemble 208 to be trained. Generally, a model refers to a computer-implemented mathematical or statistical model that, after training, can output a numerical or categorical prediction based on input data. A model ensemble comprises a collection of models. Within the model ensemble, each machine learning model can be trained separately based on a subset of a dataset. The trained model ensemble is used to generate a collection of model predictions. For example, the untrained model ensemble 208 can include a plurality of (e.g., 1,000) neural networks to be trained. As another example, the untrained model ensemble 208 can include a plurality of support vector machine regression models, Gaussian process models, radial basis function networks, partial-least-squares, mechanistic- or physical process models, generalized linear

models, support vector machine, nearest neighbor, random forest, neural networks, or any combination thereof.

[0073] In some embodiments, model evaluation techniques such as cross-validation are used to compare different classes of machine learning model, select features to be included in a particular model, perform regularization of the model, and tune model hyper-parameters. Features generally refer to variables that are derived through mathematical operations from directly observed or known quantities from which the predictions of a machine learning model are derived. Examples of features include any or all of the media and physical process conditions which are derived by identity, or more complex derivations, such as nitrogen concentration (which is derived from the composition of the media components and their concentration) and oxygenation rate (which can be derived from the sparge rate, agitation speed, and vessel size). Exemplary features can further include features from a time-course such as the maximum or minimum value during a time period, the slope during a time period, duration of specific features like the duration until a certain value is reached, the duration in which a value stays within a certain range, a calculated lag phase (e.g., oxygen sensor measurements), or any combinations thereof. Features can be manually engineered using existing process understanding and data-mining of compiled dataset. Features can also be selected using automated feature selection techniques.

[0074] In some embodiments, regularization of the models in the model ensemble is performed. Regularization refers to a mathematical term added to the loss function of a machine learning model in order to impose additional penalty on parameter magnitudes during model training. For example, l2- or ridge regression or l1- or the lasso procedure will reduce or eliminate the influence of insensitive inputs to avoid overfitting.

[0075] In some embodiments, hyper-parameters of the models in the model ensemble are tuned. Hyper-parameters generally refer to parameters of a machine learning model that may or may not be directly fit during model training. Hyper-parameters can include variables related to the machine learning model or to the algorithm used to train the machine learning model. For example, the size of the hidden layers of the neural network is a model hyper-parameter, whereas the seed for the random number generator or loss function are training hyper-parameters.

[0076] In some embodiments, evaluation metrics, such as model error rates, are used to decide an appropriate size of the machine learning model ensemble (e.g., number of neural networks).

[0077] At block 210, the system trains the model ensemble 208 using the trained data 206. In some embodiments, at block 211, the training data 206 is subsampled using resampling techniques, such as bootstrap or jackknife, which can produce an ensemble of datasets subsampled from the original training data 206. Resampling refers to a method of generating new datasets by randomly subsampling data points with replacement from an existing dataset. The datasets in the ensemble of datasets are used to train the models in the model ensemble 208.

[0078] **FIG. 3** illustrates an exemplary process 300 for training a model ensemble based on training data, according to various embodiments. As shown, training data 302 (e.g., training data 206 of **FIG. 2**) is randomly resampled into N datasets: Random Data Subsample 1, Random Data Subsample 2, Random Data Subsample 3, ..., and Random Data Subsample N. Any resampling technique can be used to resample the training data 302 into the N subsampled datasets, such as bootstrap resampling techniques.

[0079] The N datasets are used to train the N models in a model ensemble (e.g., model ensemble 208 of **FIG. 2**) to obtain N trained models: Trained Models 1, 2, 3, ..., and N. Model training refers to the process whereby the parameters of a mathematical model are algorithmically adjusted such that the model's performance on the training data is optimized with respect to a loss function, which is a mathematical equation defining the penalties applied to a machine learning model due to deviations between the quantities predicted by the model and the actual quantities observed in the training data.

[0080] In some embodiments, the mathematical models in the model ensemble are neural networks. As such, the N data subsets are used to train N untrained neural networks in a model ensemble. Because the N data subsets may be different from each other and the N untrained neural networks may be different from each other, the resulting N trained neural networks may be different from each other, for example, in terms of architecture (e.g., number of layers) and parameters.

[0081] Turning back to **FIG. 2**, the system obtains trained model ensemble 212 after block 210. For example, the trained model ensemble 212 can include a collection of trained neural networks. Each trained neural network can, based on input data (e.g.,

fermentation conditions), provide a prediction output (e.g., fermentation outcome such as yield). As discussed above, the trained neural networks in the trained model ensemble 212 may vary in their architectures and parameters.

[0082] At block 214, the system identifies a collection of optimal solutions based on the trained model ensemble 212. As shown in **FIG. 3**, the system uses the N trained models to derive N optimal solutions. Specifically, each of the N trained models in the ensemble is subjected to a mathematical or computational optimization procedure in order to derive the optimal model inputs (e.g., optimal fermentation conditions such as optimal amounts of media ingredients, optimal physical conditions). The optimization procedure comprises optimizing (e.g., minimizing, maximizing) an objective function, as described below.

[0083] The optimization procedure can include any optimization algorithm or search heuristics outputting a set of fermentation conditions.

[0084] In some embodiments, the algorithm is population-based such as gradient based optimization coupled to multi-start, genetic algorithm, scatter search, particle swarm optimization, or any combination thereof.

[0085] In some embodiments, the optimization procedure is a genetic algorithm. Generally, a genetic algorithm is a method for solving optimization problems. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individual solutions from the current population to be parents by their fitness. “Fitness” is a chosen optimality criterion and is also called an objective function which may include cost optimization, fermentation outcome optimization, or more complicated optimization rules than mere maximization. The selected parents are used to produce the children for the next generation by recombination of parameters of individual solutions and mutations (i.e., applying changes) to individual solutions. Over successive generations, the population “evolves” toward optimal solution(s) for a particular model and fitness function. In some embodiments, the algorithm does not converge to a single solution, but rather allows for multiple optimal solutions to be brought forward for each model.

[0086] **FIGS. 4A-B** illustrates an exemplary process for using a genetic algorithm to obtain optimal solution(s) based on a trained neural network, according to various embodiments. As shown in **FIG. 4A**, an exemplary trained neural network can receive input data (e.g., fermentation process conditions such as media ingredients and physical conditions)

and output a predicted fermentation outcome (e.g., yield). This trained neural network can represent any one of the trained neural networks in the trained model ensemble 212 (**FIG. 2**) or any of the trained models 1-N (**FIG. 3**).

[0087] As shown in **FIG. 4B**, a genetic algorithm is used to solve for model
5 inverse, that is, to solve for a set of optimal model inputs (e.g., an optimal set of fermentation conditions) that would produce an optimized objective function. The objective function can be calculated from the fermentation outcomes, such as formed product (e.g., biological chemistry, biomolecules, proteins with a function of interest, whole cells, metabolic products (metabolites), biotransformation products), spores, colony forming units (CFU), packed cell
10 volume (PCV), degree of sporulation, viscosity, measures for shelf-life, or biological assay performance (e.g., field trial, in vitro assay, etc), viscosity, attributes of the product or product mixtures (spore fitness, post-translational modifications, protein folding, aggregation state of protein, chiral properties, alternative spliced versions, product mixture fractions), concentration or amount of by-products or impurities (host-cell protein, residual media components,
15 precipitated compounds). Additionally or alternatively, the objective function can be calculated based on costs (for materials, media ingredients and operation). As an example, the objective function can be equal to the expected revenue from the amount of formed product minus the costs for goods and operation.

[0088] The genetic algorithm starts with a population of solutions (e.g., candidate
20 sets of fermentation conditions) and repeatedly modifies the population of solutions over multiple steps. At each step, the genetic algorithm evaluate the solutions (e.g., sets of fermentation conditions) based on the objective function (e.g., calculated based on the corresponding neural network outputs, which are fermentation outcomes, and based on costs) and selects solutions from the current population to be parents. The parents are used to
25 produce the children to be included in the next population by recombination (mixing between one or several solutions) and mutation (alterations of one solution) of their solution parameters. Over successive generations, the population may evolve toward an optimal solution (e.g., an optimal set of fermentation conditions) for a given machine learning model. The genetic algorithm terminates when convergence occurs (e.g., the population is no longer
30 changing substantially) or when a maximum number of iterations are reached.

[0089] In some embodiments, the genetic algorithm requires user defined inputs related the mathematical objective to be optimized (i.e., objective function or optimality

criterion) and any constraints related to the model inputs. As discussed above, an objective function refers to a mathematical description of the quantity or quantities to be optimized along with any solution constraints. For example, an objective function might be the total amount of active ingredient produced divided by the quantity amounts of the chemical inputs multiplied by their approximate cost; this function might require positive concentrations as constraints. A constraint for a model input can be a permissible range of values, where values falling outside the range are undesirable, unrealistic, or both. For example, a constraint for ammonium sulfate can be set to be (0, 5 g/L). As another example, a constraint for run time can be set to be (120 hours, 120 hours), thus fixing the fermentation process to 5 days. A constraint for more than one input can be a limit to any linear combination of inputs or any nonlinear function of inputs. As an example, two times the level of Na_2CO_3 plus the level of NaCl must be in-between 30 and 40 to limit the overall sodium content.

[0090] Thus, as shown in the example depicted in **FIG. 3**, the system can use N implementations of genetic algorithms on the N trained models to obtain N optimal solutions. The N optimal solutions can be N sets of optimal fermentation conditions. In some embodiments, the N implementations of genetic algorithms are initialized differently (e.g., different parameters, different populations, different objective functions or different constraints) to prevent the N implementations from converging to the same/similar optimal solutions. In some embodiments, the N implementations of genetic algorithms have the same constraints and termination conditions.

[0091] Turning back to **FIG. 2**, the system obtains an ensemble of optimized solutions 216. As discussed with reference to **FIGS. 3** and **4A-B**, the ensemble of optimized solutions 216 may be a collection of sets of fermentation conditions, wherein each set produces an optimal output (e.g., optimal yield, optimal cost, or a combination thereof) given a trained machine learning model in the model ensemble 212.

[0092] At block 218, the system evaluates the ensemble of optimized solutions 216. Evaluating the optimized solutions 216 can include deriving statistical confidence in the predictions and to uncover new process insights through multivariate statistical analysis and data mining, as discussed below with reference to **FIG. 5**.

[0093] **FIG. 5** illustrates an exemplary process 500 for evaluating an ensemble of optimized solutions, according to various embodiments. The ensemble of optimized solutions 502 can be the ensemble of optimized solutions 216 (**FIG. 2**) or the optimal solutions 1-N

(**FIG. 3**). Process 500 is performed, for example, using one or more electronic devices implementing a software platform. In some examples, process 500 is performed using a client-server system, and the blocks of process 500 are divided up in any manner between the server and client device(s). Thus, while portions of process 500 are described herein as being performed by a particular electronic device, it will be appreciated that process 500 is not so limited. In process 500, some blocks are, optionally, combined, the order of some blocks is, optionally, changed, and some blocks are, optionally, omitted. In some examples, additional steps may be performed in combination with the process 500. Accordingly, the operations as illustrated (and described in greater detail below) are exemplary by nature and, as such, should not be viewed as limiting.

[0094] At block 504, the system (e.g., one or more electronic devices) clusters the ensemble of optimized solutions. Clustering refers to a method used to explore underlying structure in a dataset by assigning certain data points in the dataset to a cluster based on their underlying mathematical similarity to each other. Different trained models in the trained model ensemble (e.g., trained model ensemble 216 in **FIG. 2**, trained models 1-N in **FIG. 3**) may lead to similar optimal solutions (e.g., sets of fermentation conditions). Thus, clustering can be used to group similar optimal solutions.

[0095] As shown in **FIGS. 6A-B**, plotting the ensemble of optimal solutions into principal components space shows clear clustering of the optimal solutions. **FIG. 6A** illustrates a 2-dimensional projection of an exemplary ensemble of optimal solutions along the PC1 axis and the PC2 axis. **FIG. 6B** illustrates a 2-dimensional projection of the same ensemble of optimal solutions along the PC2 axis and the PC3 axis. The clustering demonstrates convergence of different machine learning models in the ensemble to similar optimal solutions.

[0096] Optionally, the system performs multivariate statistical analysis within and between clusters and identifies key process variables, respectively. As discussed below, multivariate statistical analysis of the solution ensemble results in new knowledge into the key features in the process as well as limitations of the training data.

[0097] Methods for multivariate analysis can include any computational, mathematical, or statistical procedure aimed at identifying correlations in the solution ensemble. Specific examples include linear decomposition such as partial least squares (PLS) and principal component analysis (PCA) along with other exploratory data analysis or data

mining procedures. Data mining can include any combination of computational, statistical, and mathematical techniques used to uncover new knowledge about the statistical associations and processes underlying the generation of a dataset.

[0098] **FIGS. 7A-B** illustrates plotted PCA loadings of an exemplary ensemble of optimal solutions. The PCA loadings can be used to examine diversity of the optimal solutions and input parameter correlation with diversity. As shown, each of the principal components is dominated by a few variables, indicating that variability in the ensemble solutions is largely due to variation in a few process variables. This analysis also demonstrates the ability to obtain new knowledge of the underlying process using the ensemble approach. Factors that vary between models in the ensemble indicate either interactions between variables or poorly resolved effects of a variable. Further experimentation can be used to validate the interactions between variables or resolve the effects of a variable. Alternatively a poorly resolved effect can indicate a low sensitivity of the objective function to this variable and further exploration of the effect of those variables can be omitted. In contrast, factors that do not vary result from high concordance among models in the ensemble.

[0099] In this particular example, the influence of the process conditions on the objective function “yield” is shown. High loadings of Manganese in PC1 indicates that altering the concentrations of manganese will not have a significant impact on the yield parameters, as Manganese has the most variation in concentrations that result in optimal yield/objective value.

[00100] As discussed above, at block 504, the ensemble of optimized solutions are clustered. The solutions can be clustered using any type of algorithm or procedure that groups together the solutions based on a quantitative metric of similarity. This includes algorithms such as k-means clustering, x-means clustering, and hierarchical clustering. An elbow curve can be used to determine the number of clusters (k). As shown in **FIG. 8**, an exemplary elbow curve shows the score of the k-means clustering and indicates that increase in performance diminishes after around 10-15 distinct clusters. In some examples, silhouette coefficient (or score) can be used instead.

[00101] Solutions belonging to a cluster can be summarized using any method that averages over variation in the cluster, including using cluster centroids. This cluster summary is also referred to as a “solution prototype” or “design prototype”. In other words, after

clustering the ensemble of optimized solutions, the system can obtain a plurality of solution prototypes corresponding to the plurality of clusters identified. Each solution prototype corresponds to a set of averaged optimal fermentation conditions for a cluster. For each solution prototype, the size of the corresponding solution cluster relative to the overall size of the ensemble can be assumed to relate to the statistical confidence in that solution prototype. For example, solution clusters with smaller numbers of associated solutions can be associated with a lower confidence score, as the number of training sets that converged to that solution is smaller.

[00102] At block 510, the system makes model ensemble predictions based on the solution prototypes. In some embodiments, the system can run each solution prototype through the trained model ensemble (e.g., trained model ensemble 212 of **FIG. 2**) to generate model predictions (e.g., predictions of fermentation outcome such as yield). The resulting model predictions provides a probability function for the yield for a fermentation process, such that e.g., 99% or 95% quantile for the yield at given fermentation conditions can be obtained. For each solution prototype, there is a corresponding cost (in time, material, labor, operation). Thus, this provides insights into whether a particular yield goal is desirable and the design prototypes that can produce the yield (as well as the associated costs).

[00103] The system can also generate bootstrap distributions of the model predictions. The calculated yield and prediction intervals of each proposed solution cluster can be plotted to demonstrate diversity of model solutions. At block 512, numerical prediction intervals can be estimated for each solution of interest by generating a distribution of model predictions using the model ensemble. Quantiles of this distribution are subsequently used to define cutoffs for any defined confidence/prediction limit.

[00104] **FIG. 9** illustrates 95% prediction intervals of a set of exemplary solution prototypes and the extent of convergence among a model ensemble, according to various embodiments. Each bar indicates the upper and lower 95% quantiles for the solution prototype. The size of dots is proportional to the number of models in the ensemble supporting the predicting each candidate solutions, giving an estimate of the overall ensemble support for each solution prototype. Thus, the ensemble approach allows for the estimation of upper and lower prediction interval for each solution prototype.

[00105] At block 514, the system performs experimental design and empirical validation. A new set of experiments for process development can be generated based on the

numerical solution prototypes generated by clustering in block 504, numerical prediction intervals obtained in blocks 510 and 512. For example, the prediction intervals and size of the clusters are used to judge prediction quality and prioritize obtained fermentation design prototypes according to their importance. Integration of the information can be done using
5 either automated computational pipelines or manually using knowledge derived from the analysis and existing subject matter expertise, resulting in a set of new conditions and/or process variables for further exploration. For example, the solution prototypes might indicate that manganese concentrations of optimal yield varied greatly between solutions and furthermore inversely correlated with iron, indicating that additional experiments should be
10 performed to explore the interaction of these two fermentation parameters. Experiments can be designed using any numerical, statistical, or computational procedure for generating new experimental conditions including traditional experimental design methods.

[00106] Turning back to **FIG. 2**, after the system performs experimental design and/or empirical validation, the system obtains new empirical data to be used as new training
15 data 222. The new data can be used to retrain the machine learning model ensemble, which can subsequently be used as input to block 214 resulting in a cycle of continual process optimization. This cycle (blocks 210-220) can be repeated as many times as needed in order to reach process optimization objectives. Various methods can be used in order to retrain the models, including methods which utilize boosting to increase model performance on newly
20 derived data. Boosting refers to methods of training a machine learning model that give disproportionate weights to a subset of the training dataset in order to enhance model performance on that subset. Alternatively, new process data can be used as input to block 210 and the entire process can be repeated.

[00107] **FIG. 10** illustrates an exemplary computer-enabled method for optimizing a
25 fermentation process, according to various embodiments. Process 1000 is performed, for example, using one or more electronic devices implementing a software platform. In some examples, process 1000 is performed using a client-server system, and the blocks of process 1000 are divided up in any manner between the server and client device(s). Thus, while portions of process 1000 are described herein as being performed by a particular device, it will
30 be appreciated that process 1000 is not so limited. In process 1000, some blocks are, optionally, combined, the order of some blocks is, optionally, changed, and some blocks are, optionally, omitted. In some examples, additional steps may be performed in combination

with the process 1000. Accordingly, the operations as illustrated (and described in greater detail below) are exemplary by nature and, as such, should not be viewed as limiting.

[00108] At block 1002, a system (e.g., comprising one or more electronic devices) subsamples a training dataset to obtain a plurality of subsampled datasets, wherein the training
5 dataset includes data related to one or more previous instances of the fermentation process. At block 1004, the system trains a plurality of mathematical models based on the plurality of subsampled datasets. At block 1006, the system identifies a plurality of sets of fermentation conditions based on the plurality of trained mathematical models. At block 1008, the system identifies one or more clusters of sets of fermentation conditions based on the plurality of sets
10 of fermentation conditions. At block 1010, the system obtains one or more fermentation design prototypes based on the one or more clusters. At block 1012, the system determines a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

[00109] In some embodiments, the method further comprises performing a new
15 instance of the fermentation process based on the candidate fermentation condition.

[00110] In some embodiments, the method further comprises obtaining a new training dataset based on the new instance of the fermentation process; and training the plurality of mathematical models based on the new training dataset.

[00111] In some embodiments, the training data comprises data related to one or
20 more fermentation conditions of the fermentation process.

[00112] In some embodiments, the one or more fermentation conditions comprise one or more media ingredients, one or more physical conditions, online data, one or more process data points, or any combination thereof.

[00113] In some embodiments, the one or more media ingredients comprise: water
25 content, carbon content, nitrogen content, nitrogen sources (e.g., ammonia, urea, protein, amino acids), carbohydrates (simple (e.g., glucose, sucrose, fructose, oligosaccharides) and complex (e.g., maltodextrins, starch)), trace elements (e.g., magnesium, potassium, iron, zinc, manganese, calcium, copper, potassium, phosphorus, sodium, potassium, sulfate, chloride), vitamins (biotin, folic acid, thiamine), complex media (commercially available, soy/corn
30 flours or meals, yeast extracts, molasses, corn steep liquor, cottonseed-based flour), small organic acids (lactate, acetate), amino acids, precursors for a desired product (e.g., glutamate,

pyruvate), proteins, peptides, fat or fatty acids, buffer components (e.g., CaCO_3 , phosphate-buffers) and anti-foam, or any combination thereof.

[00114] In some embodiments, the one or more physical conditions comprise: one or more physical conditions comprise: pH, temperature, sparge rate, CO_2 flow rate, O_2 flow rate, sterilization time, sterilization temperature, addition of components after sterilization, or any combination thereof.

[00115] In some embodiments, the one or more online data comprise: measurements of pH, temperature, sparge rate, dissolved oxygen, CO_2 flow rate, O_2 flow rate, respiration-carbon evolution rate (CER), oxygen uptake rate (OUR), mixing rate, ingredient feed rates, ingredient feed times, temperature profiles, antifoam, sterilization time, sterilization temperature, total process time or any combination thereof.

[00116] In some embodiments, the one or more process data points comprise features, which are extracted from time-course data or online probe measurements (like the maximum- or minimum value, the slope, duration of specific features like the duration until a certain value is reached, the duration in which a value stays within a certain range, a calculated lag phase (e.g., from oxygen sensor measurements) or combinations thereof).

[00117] In some embodiments, the training data comprises data related to one or more fermentation outcomes of the fermentation process.

[00118] In some embodiments, one or more fermentation outcomes comprise: one or more formed products, one or more measures for characterizing biomass, one or more properties of the solution, one or more biological assay performance, one or more attributes of the product, one or more concentration or amount of by-products or impurities, one or more costs of ingredients, or any combination thereof.

[00119] In some embodiments, the one fermentation outcomes comprise: absolute amount or relative measure of the formed product (e.g., biological chemistry, biomolecules, proteins with a function of interest, whole cells, metabolic products (metabolites), biotransformation products), additional measures characterizing biomass (cell dry weight, spores count, colony forming units (CFU), packed cell volume (PCV), degree of sporulation), biological assay performance (e.g., field trial, in vitro assay, etc.), properties of the solution (viscosity), attributes of the product or product mixtures (spore fitness, post-translational modifications, protein folding, aggregation state of protein, chiral properties, alternative spliced versions, product mixture fractions, measures for shelf-life), concentration or amount

of by-products or impurities (host-cell protein, residual media components, precipitated compounds), costs (for materials, media ingredients and operation) or a combination thereof.

[00120] In some embodiments, the training data is derived from one or more previous fermentation experiments.

5 [00121] In some embodiments, subsampling the training dataset comprises subsampling the training dataset via a bootstrap resampling technique.

[00122] In some embodiments, the plurality of mathematical models are machine learning models.

[00123] In some embodiments, the plurality of machine learning models comprises
10 a plurality of neural networks.

[00124] In some embodiments, the plurality of machine learning models comprises a plurality of support vector machines.

[00125] In some embodiments, the plurality of mathematical models comprises:
one or more Gaussian process models, one or more radial basis function networks, one or
15 more partial-least-squares, generalized linear models, one or more support vector machines,
one or more nearest neighbor algorithms, one or more random forest algorithms, one or more
neural networks, or any combination thereof.

[00126] In some embodiments, the plurality of mathematical models comprises a
first mathematical model and a second mathematical model; the plurality of subsampled
20 datasets comprises a first subsampled dataset and a second subsampled dataset; and training
the plurality of mathematical models based on the plurality of subsampled datasets comprises:
training the first mathematical model based the first subsampled dataset and training the
second mathematical model based on the second subsampled dataset.

[00127] In some embodiments, each of the plurality of trained mathematical models
25 is configured to provide a predicted fermentation outcome based on a set of fermentation
conditions.

[00128] In some embodiments, identifying the plurality of sets of fermentation
conditions based on the plurality of trained mathematical models comprises: for each trained
mathematical model of the plurality of trained mathematical models, identifying a set of
30 fermentation conditions based on a optimization algorithm which maximizes or minimizes an
objective function.

[00129] In some embodiments, the optimization algorithm is a population-based search heuristics.

[00130] In some embodiments, the optimization algorithm is a genetic algorithm.

[00131] In some embodiments, the optimization algorithm is configured to identify
5 a set of fermentation conditions that optimizes the objective function.

[00132] In some embodiments, the objective function is calculated based on one or more fermentation outcomes predicted by a trained mathematical model, a cost value, or any combination thereof.

[00133] In some embodiments, an objective function comprises a user-specified
10 formula.

[00134] In some embodiments, identifying the one or more clusters comprises: performing an unsupervised clustering algorithm on the plurality of sets of fermentation conditions.

[00135] In some embodiments, identifying the one or more clusters comprises:
15 performing k-means or any other unsupervised clustering algorithm (e.g., x-means, hierarchical clustering, mixture of Gaussians) on the plurality of sets of fermentation conditions.

[00136] In some embodiments, each fermentation design prototype of the one or more fermentation design prototypes corresponds to a centroid of the corresponding cluster of
20 the one or more clusters.

[00137] In some embodiments, the method further comprises calculating a prediction intervals and a number of members associated with each fermentation design prototype based on the corresponding cluster.

[00138] In some embodiments, determining a candidate fermentation condition for
25 the fermentation process based on the one or more fermentation design prototypes comprises: selecting the candidate fermentation condition based on the prediction interval of each fermentation design prototype and the number of members of the associated cluster.

[00139] In some embodiments, the method further comprises: performing a multivariate analysis based on the plurality of sets of fermentation conditions to understand
30 correlations and important or not important key variables in the identified fermentation conditions.

[00140] In some embodiments, an exemplary system comprises: one or more processors; a memory; and one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for: subsampling a training dataset to obtain a plurality
5 of subsampled datasets, wherein the training dataset includes data related to one or more previous instances of a fermentation process; training a plurality of mathematical models based on the plurality of subsampled datasets; identifying a plurality of sets of fermentation conditions based on the plurality of trained mathematical models; identifying one or more clusters of sets of fermentation conditions based on the plurality of sets of fermentation
10 conditions; obtaining one or more fermentation design prototypes based on the one or more clusters; and determining a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

[00141] In some embodiments, an exemplary non-transitory computer-readable storage medium storing one or more programs, the one or more programs comprising
15 instructions, which when executed by one or more processors of an electronic device, cause the electronic device to: subsample a training dataset to obtain a plurality of subsampled datasets, wherein the training dataset includes data related to one or more previous instances of the fermentation process; train a plurality of mathematical models based on the plurality of subsampled datasets; identify a plurality of sets of fermentation conditions based on the
20 plurality of trained mathematical models; identify one or more clusters of sets of fermentation conditions based on the plurality of sets of fermentation conditions; obtain one or more fermentation design prototypes based on the one or more clusters; and determine a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

25 [00142] The operations described above with reference to **FIG. 10** are optionally implemented by components depicted in **FIG. 11**. It would be clear to a person having ordinary skill in the art how other processes described herein are implemented based on the components depicted in **FIG. 11**.

[00143] **FIG. 11** illustrates an example of a computing device in accordance with
30 one embodiment. Device 1100 can be a host computer connected to a network. Device 1100 can be a client computer or a server. As shown in **FIG. 11**, device 1100 can be any suitable type of microprocessor-based device, such as a personal computer, workstation, server or

handheld computing device (portable electronic device) such as a phone or tablet. The device can include, for example, one or more of processor 1110, input device 1120, output device 1130, storage 1140, and communication device 1160. Input device 1120 and output device 1130 can generally correspond to those described above, and can either be connectable or
5 integrated with the computer.

[00144] Input device 1120 can be any suitable device that provides input, such as a touch screen, keyboard or keypad, mouse, or voice-recognition device. Output device 1130 can be any suitable device that provides output, such as a touch screen, haptics device, or speaker.

10 [00145] Storage 1140 can be any suitable device that provides storage, such as an electrical, magnetic or optical memory including a RAM, cache, hard drive, or removable storage disk. Communication device 1160 can include any suitable device capable of transmitting and receiving signals over a network, such as a network interface chip or device. The components of the computer can be connected in any suitable manner, such as via a
15 physical bus or wirelessly.

[00146] Software 1150, which can be stored in storage 1140 and executed by processor 1110, can include, for example, the programming that embodies the functionality of the present disclosure (e.g., as embodied in the devices as described above).

[00147] Software 1150 can also be stored and/or transported within any non-
20 transitory computer-readable storage medium for use by or in connection with an instruction execution system, apparatus, or device, such as those described above, that can fetch instructions associated with the software from the instruction execution system, apparatus, or device and execute the instructions. In the context of this disclosure, a computer-readable storage medium can be any medium, such as storage 1140, that can contain or store
25 programming for use by or in connection with an instruction execution system, apparatus, or device.

[00148] Software 1150 can also be propagated within any transport medium for use by or in connection with an instruction execution system, apparatus, or device, such as those described above, that can fetch instructions associated with the software from the instruction
30 execution system, apparatus, or device and execute the instructions. In the context of this disclosure, a transport medium can be any medium that can communicate, propagate or transport programming for use by or in connection with an instruction execution system,

apparatus, or device. The transport readable medium can include, but is not limited to, an electronic, magnetic, optical, electromagnetic or infrared wired or wireless propagation medium.

5 [00149] Device 1100 may be connected to a network, which can be any suitable type of interconnected communication system. The network can implement any suitable communications protocol and can be secured by any suitable security protocol. The network can comprise network links of any suitable arrangement that can implement the transmission and reception of network signals, such as wireless network connections, T1 or T3 lines, cable networks, DSL, or telephone lines.

10 [00150] Device 1100 can implement any operating system suitable for operating on the network. Software 1150 can be written in any suitable programming language, such as C, C++, Java or Python. In various embodiments, application software embodying the functionality of the present disclosure can be deployed in different configurations, such as in a client/server arrangement or through a Web browser as a Web-based application or Web
15 service, for example.

[00151] Although the disclosure and examples have been fully described with reference to the accompanying figures, it is to be noted that various changes and modifications will become apparent to those skilled in the art. Such changes and modifications are to be understood as being included within the scope of the disclosure and examples as defined by
20 the claims.

[00152] The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments
25 were chosen and described in order to best explain the principles of the techniques and their practical applications. Others skilled in the art are thereby enabled to best utilize the techniques and various embodiments with various modifications as are suited to the particular use contemplated.

CLAIMS

1. A computer-enabled method for optimizing a fermentation process, the method comprising:

subsampling a training dataset to obtain a plurality of subsampled datasets, wherein the
5 training dataset includes data related to one or more previous instances of the fermentation process;

training a plurality of mathematical models based on the plurality of subsampled datasets;

identifying a plurality of sets of fermentation conditions based on the plurality of
10 trained mathematical models;

identifying one or more clusters of sets of fermentation conditions based on the plurality of sets of fermentation conditions;

obtaining one or more fermentation design prototypes based on the one or more clusters; and

15 determining a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

2. The method according to claim 1, further comprising: performing a new instance of the fermentation process based on the candidate fermentation condition.

20

3. The method according to claim 1-2, further comprising: obtaining a new training dataset based on the new instance of the fermentation process; and training the plurality of mathematical models based on the new training dataset.

25 4. The method according to any of claims 1-3, wherein the training data comprises data related to one or more fermentation conditions of the fermentation process.

5. The method according to claim 4, wherein the one or more fermentation conditions comprise one or more media ingredients, one or more physical conditions, online
30 data, one or more process data points, or any combination thereof.

6. The method according to any of claims 1-5, wherein the training data comprises data related to one or more fermentation outcomes of the fermentation process.

7. The method according to claim 6, wherein one or more fermentation outcomes
5 comprise: one or more formed products, one or more measures for characterizing biomass, one or more properties of the solution, one or more biological assay performance, one or more attributes of the product, one or more concentration or amount of by-products or impurities, one or more costs of ingredients, or any combination thereof.

10 8. The method according to any of claims 1-7, wherein the training data is derived from one or more previous fermentation experiments.

9. The method according any of claims 1-8, wherein subsampling the training
dataset comprises subsampling the training dataset via a bootstrap resampling technique.
15

10. The method according to any of claims 1-9, wherein the plurality of mathematical models are machine learning models.

11. The method according to claim 10, wherein the plurality of machine learning
20 models comprises a plurality of neural networks.

12. The method according to claim 10, wherein the plurality of machine learning models comprises a plurality of support vector machines.

13. The method according to claim 10, wherein the plurality of mathematical
25 models comprises: one or more Gaussian process models, one or more radial basis function networks, one or more partial-least-squares, generalized linear models, one or more support vector machines, one or more nearest neighbor algorithms, one or more random forest algorithms, one or more neural networks, or any combination thereof.

30

14. The method according to any of claims 1-13,

wherein the plurality of mathematical models comprises a first mathematical model and a second mathematical model;

wherein the plurality of subsampled datasets comprises a first subsampled dataset and a second subsampled dataset; and

5 wherein training the plurality of mathematical models based on the plurality of subsampled datasets comprises: training the first mathematical model based the first subsampled dataset and training the second mathematical model based on the second subsampled dataset.

10 15. The method according to any of claims 1-14, wherein each of the plurality of trained mathematical models is configured to provide a predicted fermentation outcome based on a set of fermentation conditions.

15 16. The method according to any of claims 1-15, wherein identifying the plurality of sets of fermentation conditions based on the plurality of trained mathematical models: for each trained mathematical model of the plurality of trained mathematical models, identifying a set of fermentation conditions based on a optimization algorithm which maximizes or minimizes an objective function.

20 17. The method according to claim 16, wherein the optimization algorithm is a population-based search heuristics.

25 18. The method according to claim 16, wherein the optimization algorithm is a genetic algorithm.

19. The method according to any of claims 16-18, wherein the optimization algorithm is configured to identify a set of fermentation conditions that optimizes the objective function.

30 20. The method according to claim 19, wherein the objective function is calculated based on one or more fermentation outcomes predicted by a trained mathematical model, a cost value, or any combination thereof.

21. The method according to any of claims 19-20, wherein an objective function comprises a user-specified formula.

5 22. The method according to any of claims 1-21, wherein identifying the one or more clusters comprises: performing an unsupervised clustering algorithm on the plurality of sets of fermentation conditions.

23. The method according to any of claims 1-22, wherein each fermentation design
10 prototype of the one or more fermentation design prototypes corresponds to a centroid of the corresponding cluster of the one or more clusters.

24. The method according to claim 23, further comprising: calculating a prediction
intervals and a number of members associated with each fermentation design prototype based
15 on the corresponding cluster.

25. The method according to claim 24, wherein determining a candidate
fermentation condition for the fermentation process based on the one or more fermentation
design prototypes comprises: selecting the candidate fermentation condition based on the
20 prediction interval of each fermentation design prototype and the number of members of the associated cluster.

26. A system, comprising: one or more processors; a memory; and one or more
programs, wherein the one or more programs are stored in the memory and configured to be
25 executed by the one or more processors, the one or more programs including instructions for:
subsampling a training dataset to obtain a plurality of subsampled datasets, wherein the
training dataset includes data related to one or more previous instances of a fermentation
process;
training a plurality of mathematical models based on the plurality of subsampled
30 datasets;
identifying a plurality of sets of fermentation conditions based on the plurality of
trained mathematical models;

identifying one or more clusters of sets of fermentation conditions based on the plurality of sets of fermentation conditions;

obtaining one or more fermentation design prototypes based on the one or more clusters; and

5 determining a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

27. A non-transitory computer-readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by one or
10 more processors of an electronic device, cause the electronic device to:

subsample a training dataset to obtain a plurality of subsampled datasets, wherein the training dataset includes data related to one or more previous instances of the fermentation process;

train a plurality of mathematical models based on the plurality of subsampled datasets;
15 identify a plurality of sets of fermentation conditions based on the plurality of trained mathematical models;

identify one or more clusters of sets of fermentation conditions based on the plurality of sets of fermentation conditions;

obtain one or more fermentation design prototypes based on the one or more clusters;
20 and

determine a candidate fermentation condition for the fermentation process based on the one or more fermentation design prototypes.

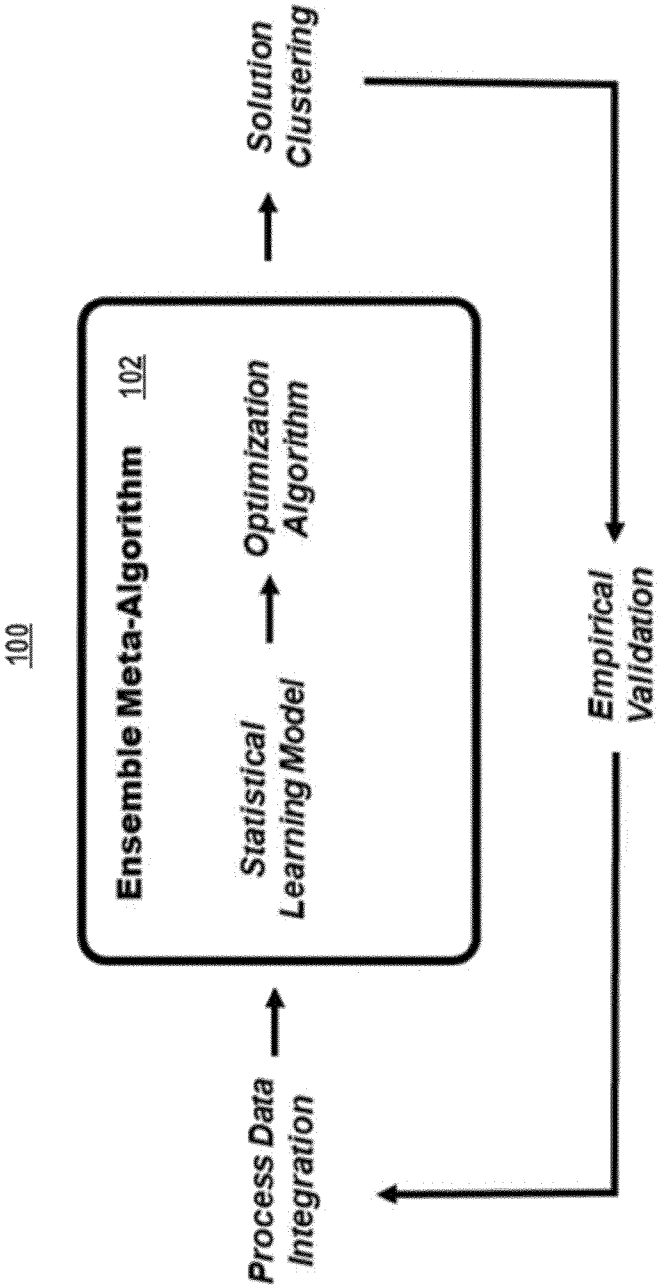
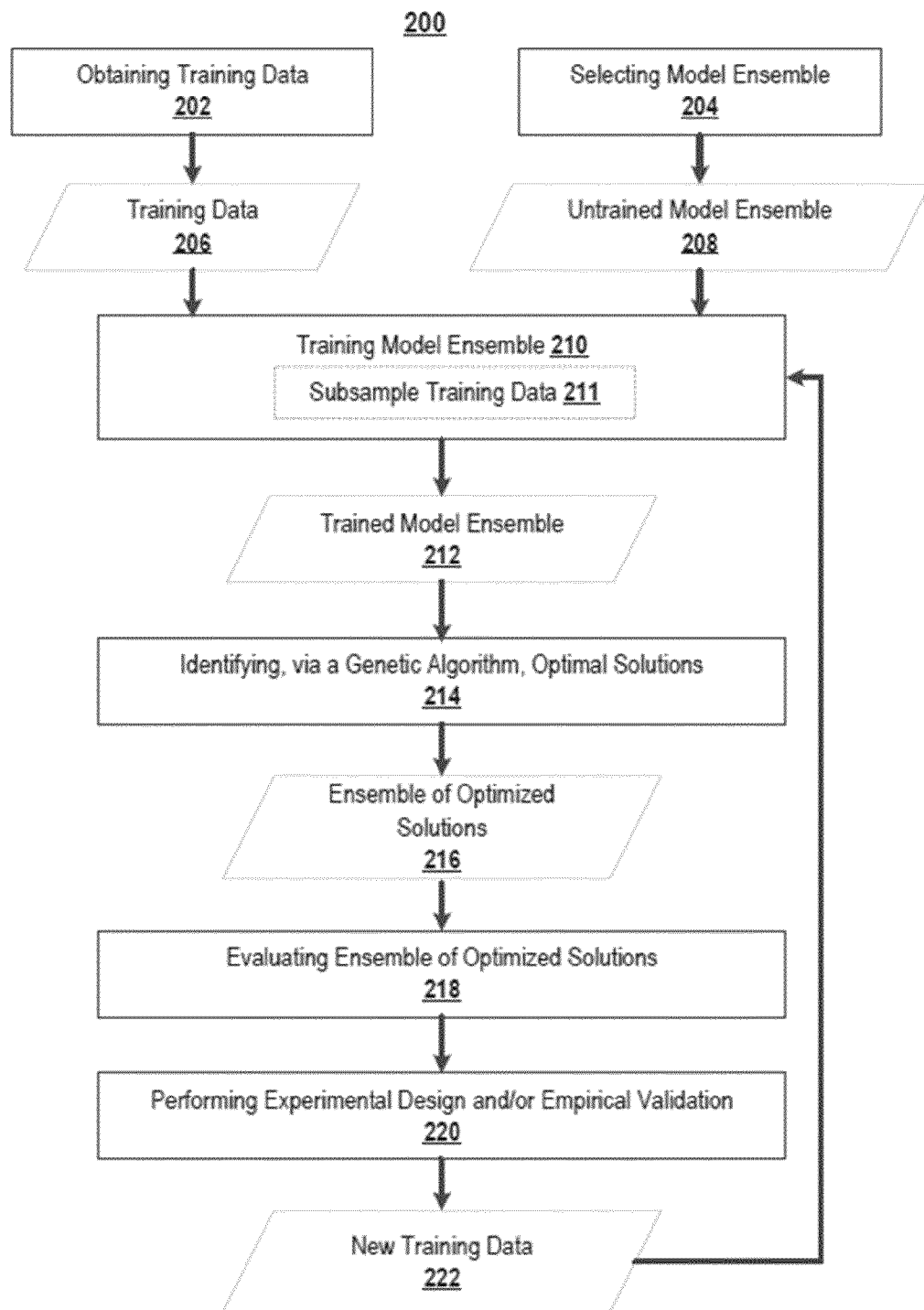


FIG. 1

**FIG. 2**

300

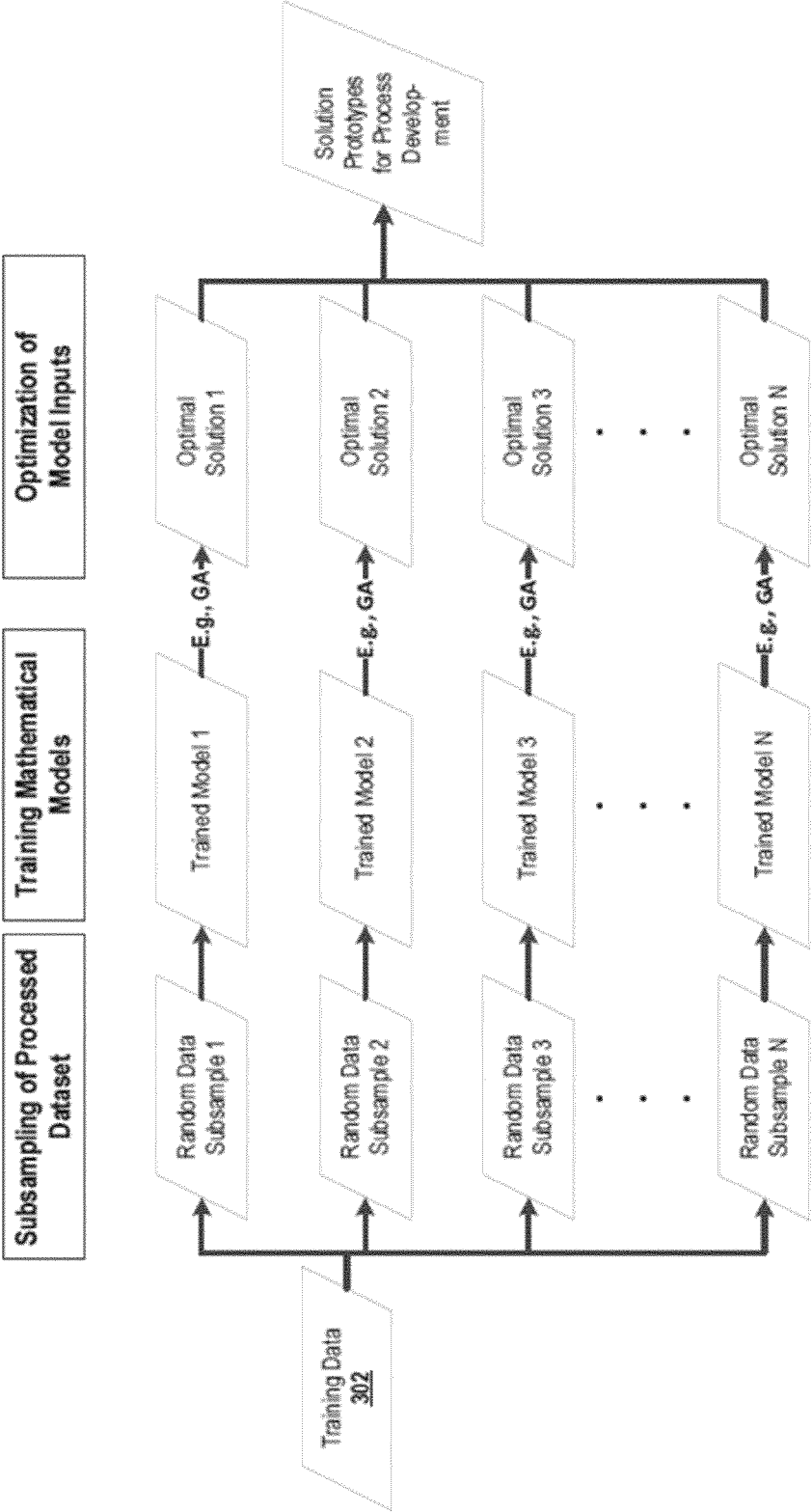


FIG. 3

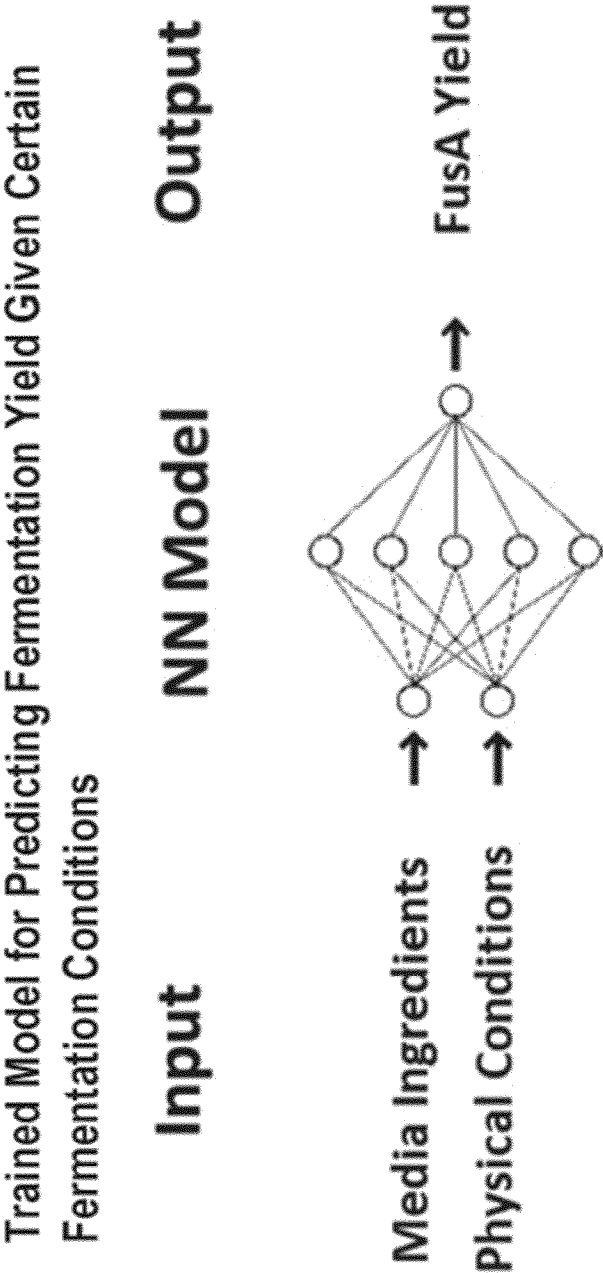


FIG. 4A

Using Genetic Algorithms to Solve for Model Inverse: Optimal Fermentation Conditions Given Trained Model

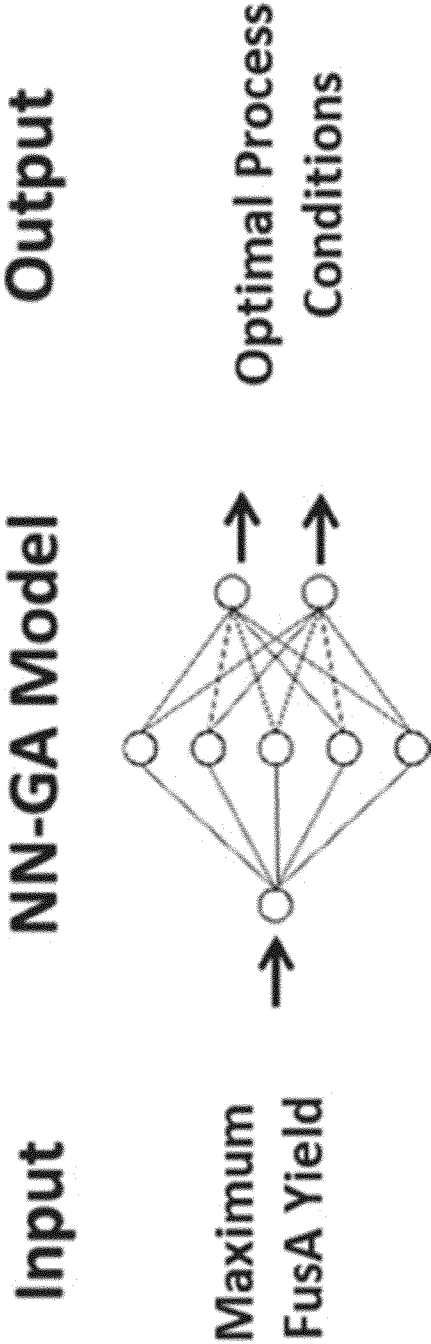


FIG. 4B

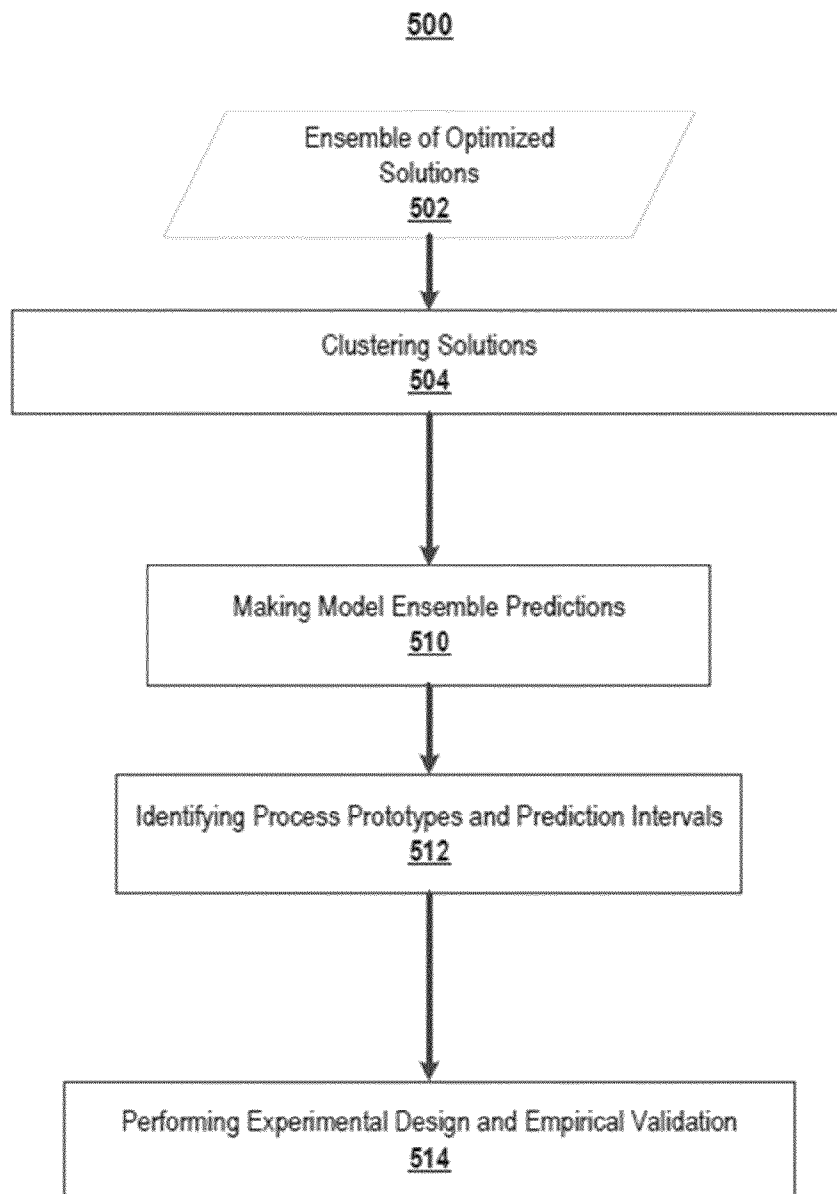


FIG. 5

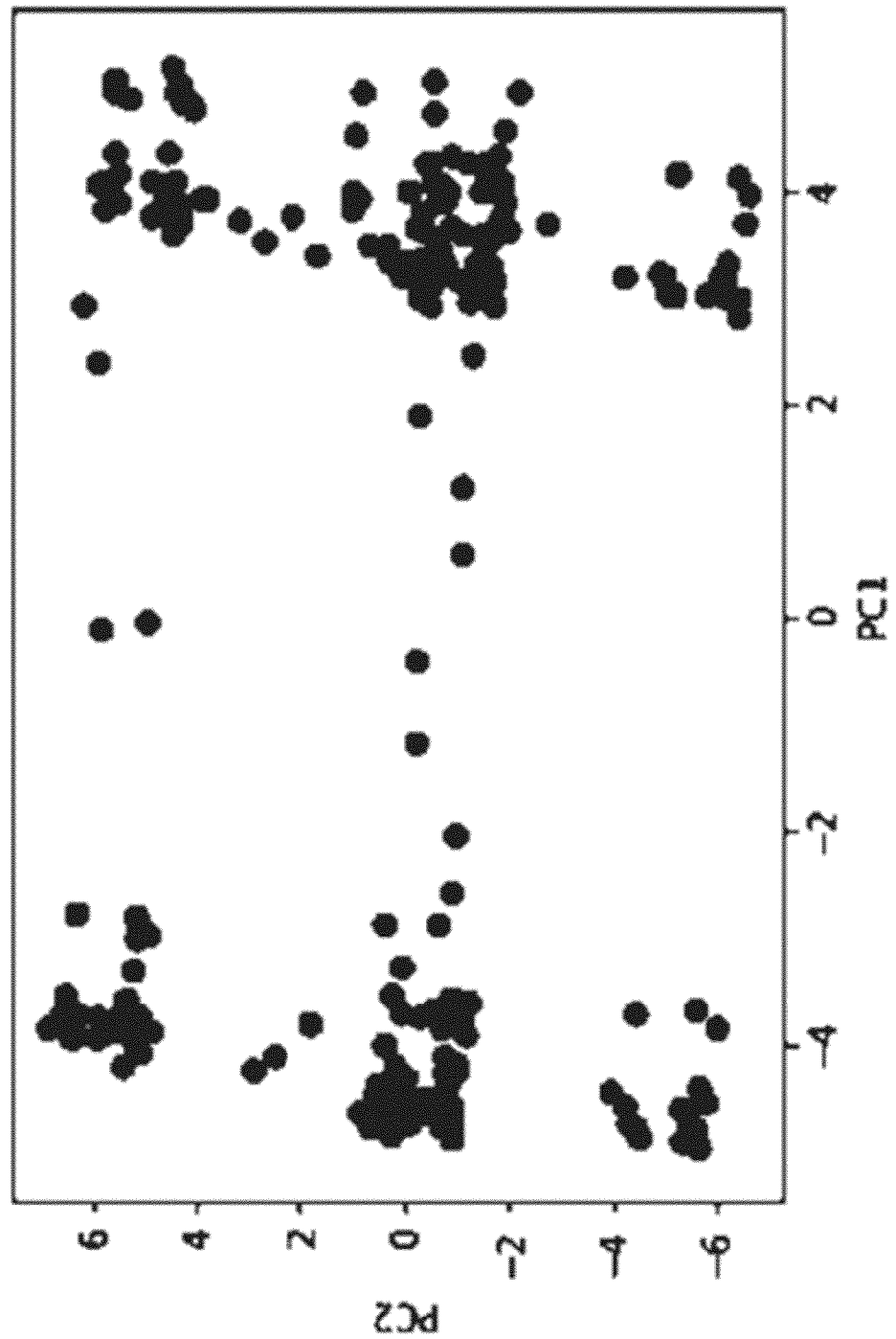


FIG. 6A

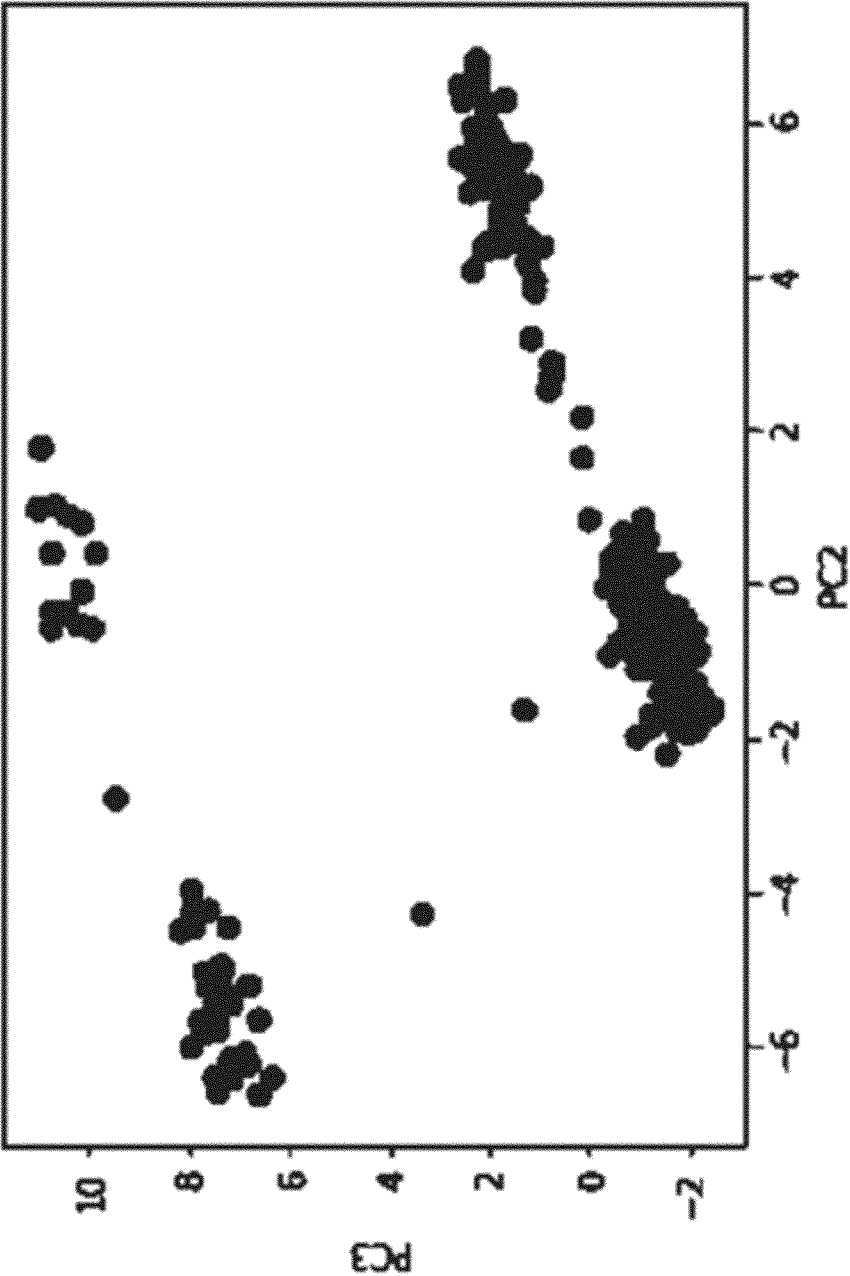


FIG. 6B

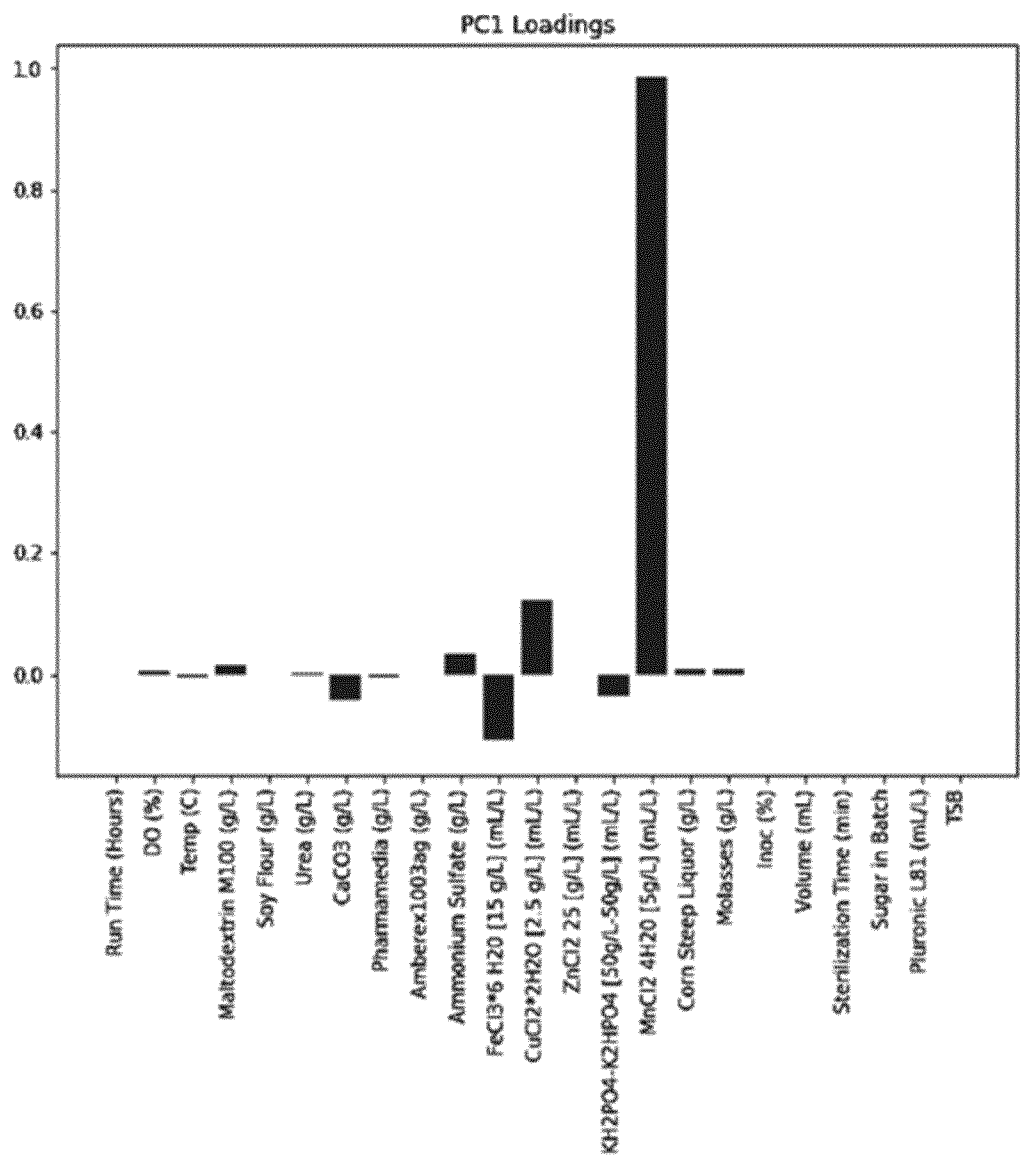


FIG. 7A

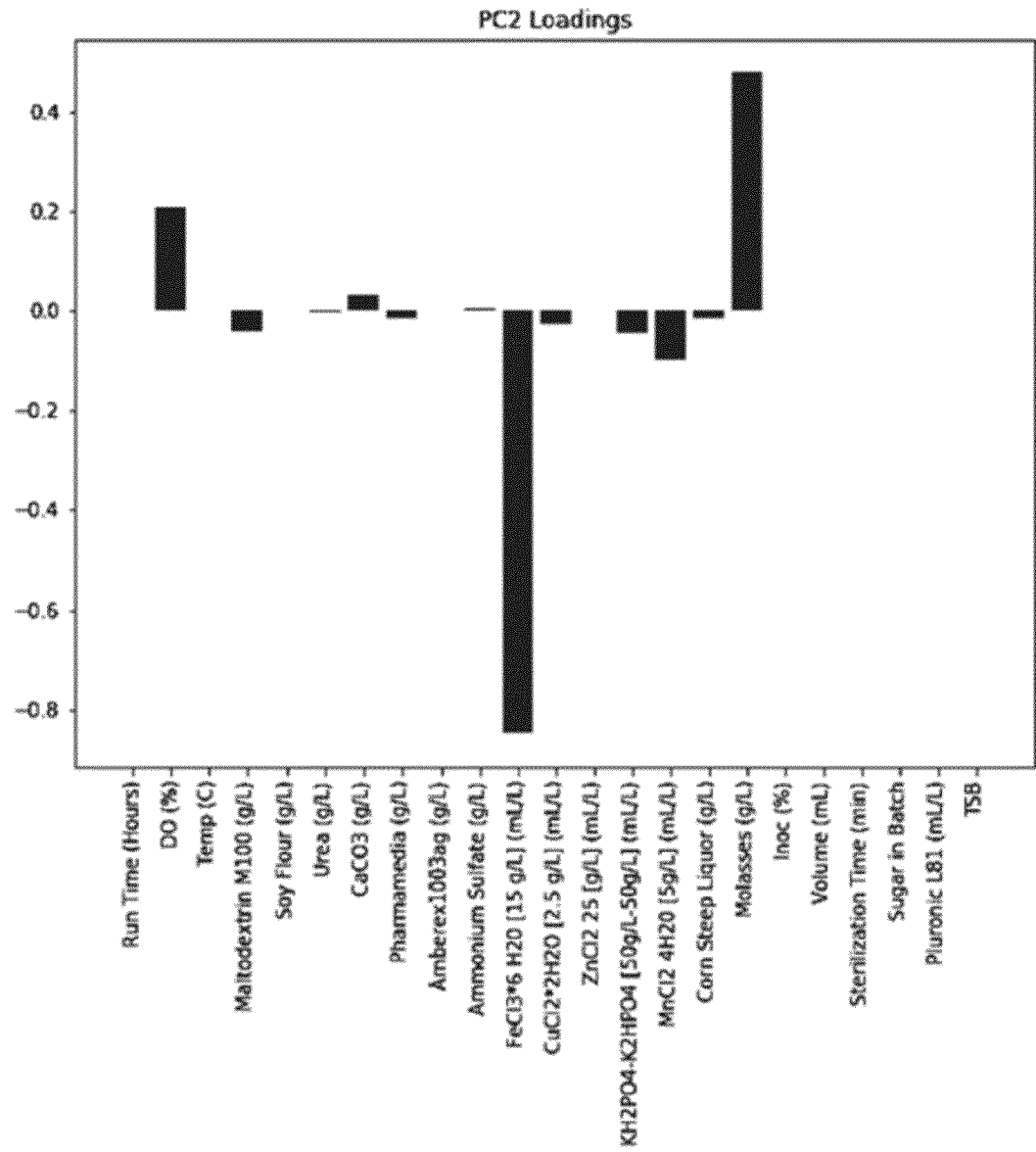


FIG. 7B

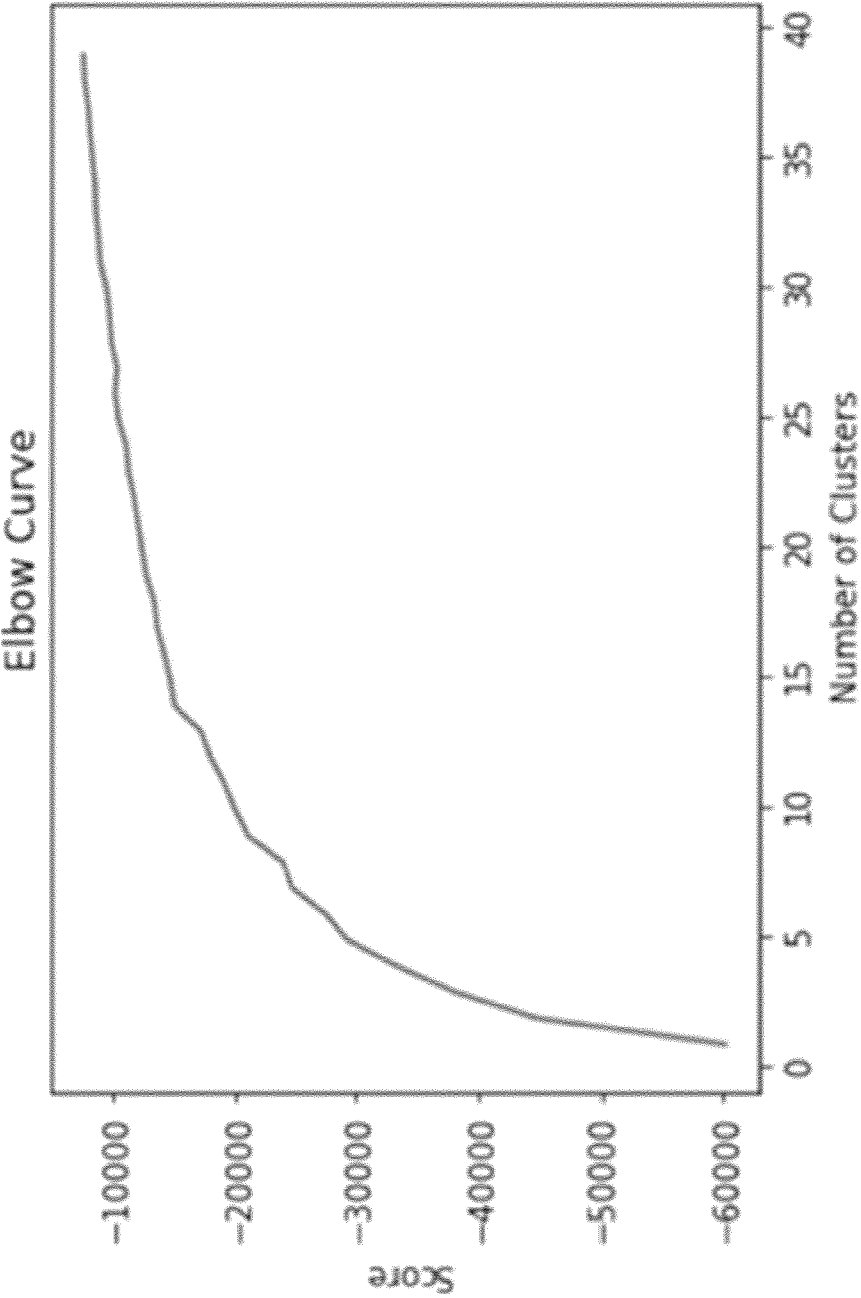


FIG. 8

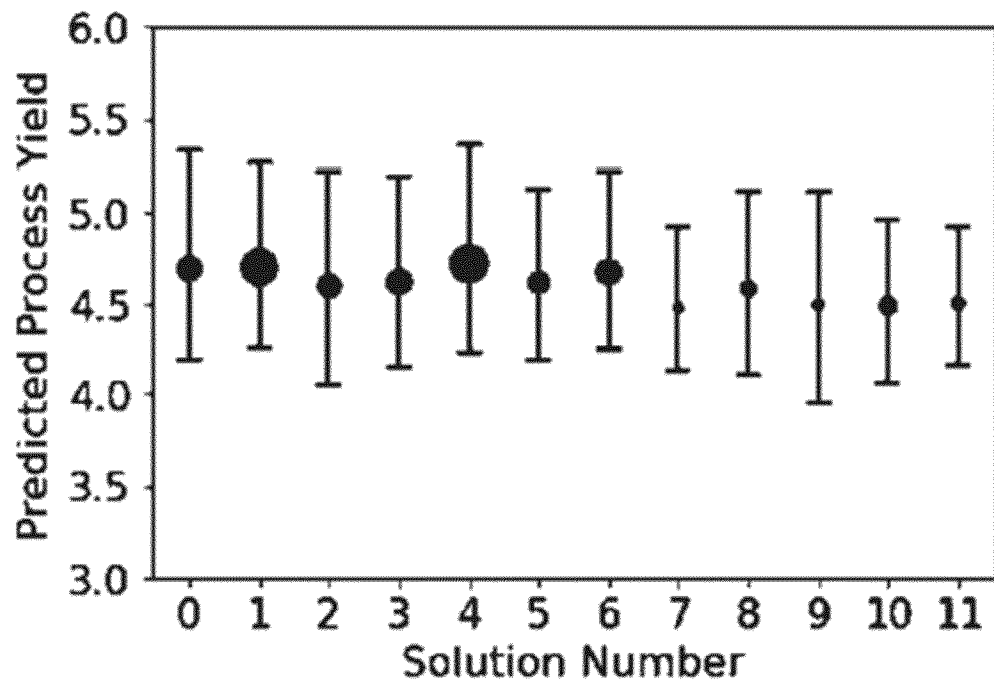


FIG. 9

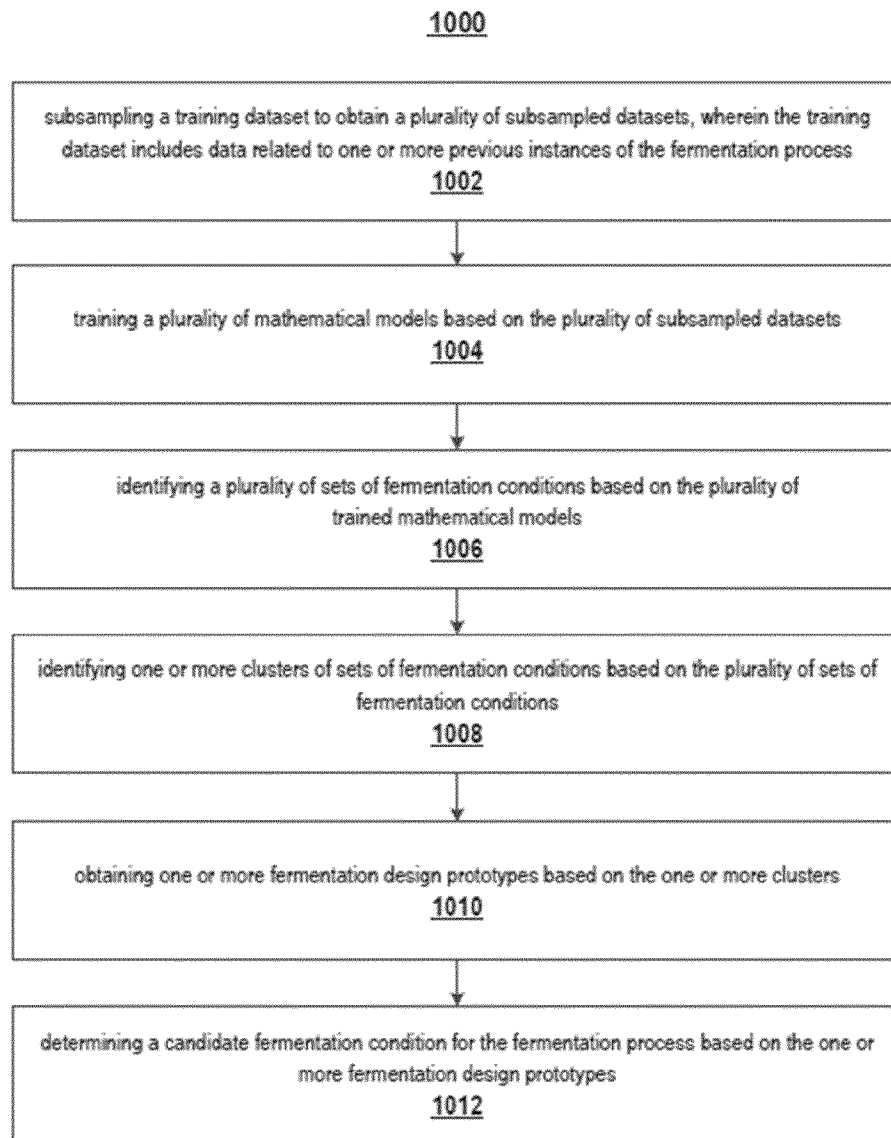


FIG. 10

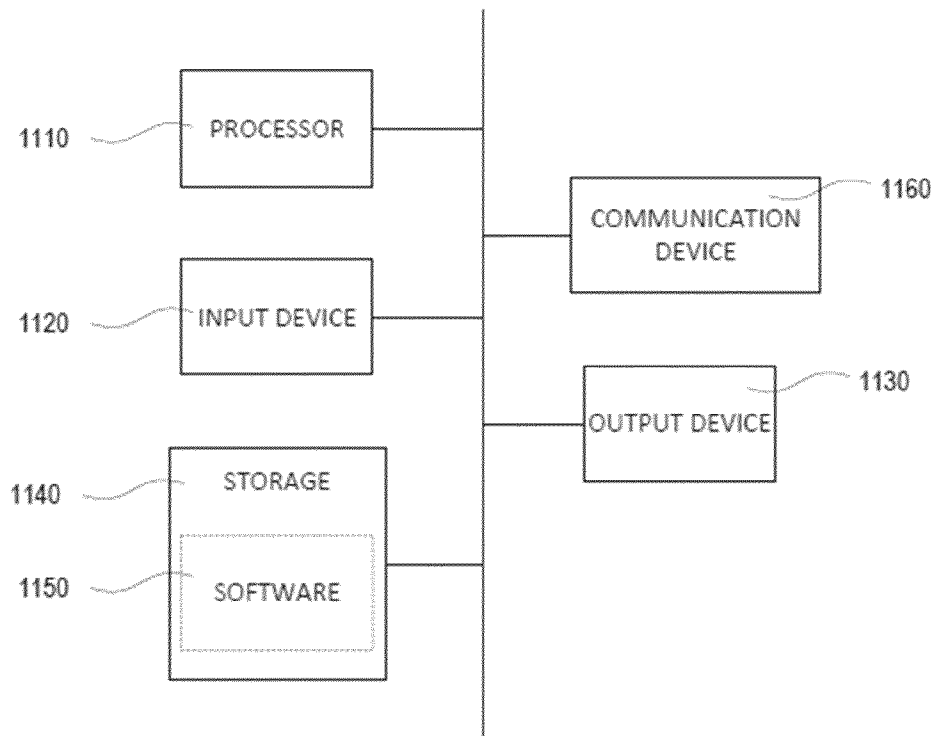


FIG. 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/053138

A. CLASSIFICATION OF SUBJECT MATTER		
INV.	G06N3/12	G06N20/20
ADD.	G06N3/00	G06N3/04
	G06N3/08	
	G06N5/00	G06N7/00
		G06N20/10
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
G06N		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
EPO-Internal, WPI Data		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ZHANG JIE ET AL: "Reliable Multi-objective On-Line Re-optimisation Control of a Fed-Batch Fermentation Process Using Bootstrap Aggregated Neural Networks", 2017 INTERNATIONAL SYMPOSIUM ON COMPUTER SCIENCE AND INTELLIGENT CONTROLS (ISCSIC), IEEE, 20 October 2017 (2017-10-20), pages 49-56, XP033321846, DOI: 10.1109/ISCSIC.2017.23 [retrieved on 2018-02-16] abstract; figures 3, 7, 8 tables I, II page 49, line 1, paragraph 2 - line 33 page 50, column 1, line 4 - line 29 page 51, column 1, line 6 - page 52, column 2, line 7 page 53, column 1, line 2 - page 54, -/--	1-27
☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
22 April 2020		06/05/2020
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Rousset, Antoine

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/053138

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	column 2, line 8 ----- Grigorios Tsoumakas ET AL: "A Taxonomy and Short Review of Ensemble Selection", 21 July 2008 (2008-07-21), pages 1-6, XP055687394, Retrieved from the Internet: URL:http://lpis.csd.auth.gr/publications/p artalas08-suema.pdf [retrieved on 2020-04-20] abstract page 2, column 2, line 38 - page 3, column 2, line 13	1-27
A	----- CN 108 268 359 A (ZHEJIANG ZHONGRUI LOW CARBON TECH CO LTD) 10 July 2018 (2018-07-10) abstract; claim 1; figure 1 -----	1-27

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2020/053138

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
CN 108268359	A	10-07-2018	NONE
