

METHOD OF OPTIMIZING AI/ML PREDICTIONS FROM A CLASSICAL DATA FEED WITH A HYBRID SIMULATOR GENERATED FROM CLASSICAL AND QUANTUM MODEL STRUCTURES

5 TECHNICAL CONTRIBUTION

The present disclosure relates to artificial intelligence / machine learning (AI/ML) simulators. More particularly, the present disclosure relates to a computer-implemented method of generating a simulator utilizing both classical model structures
10 and quantum model structures.

BACKGROUND

Simulators for AI/ML can be created from either classical model structures or,
15 alternatively, from quantum model structures. Classical simulators are configured for use of binary data, representing a single 1 or 0. Quantum simulators are configured for use of qubits as the basic building block of quantum information and typically employ wave equations to evaluate the qubit data. As discussed in “*Implementable Quantum Classifier for Nonlinear Data*” by Yuxuan Du, et al., arXiv:1809.06056v1 [quant-ph]
20 17 Sep 2018, quantum simulators can be fed with quantum data derived from classical data that has been transformed to a series of quantum inputs.

Some of the more favored model structures for classical simulators include: classical Bayesian regression trees, classical Boltzmann machine, classical restricted Boltzmann
25 machine (CRBM), classical general linear model (CGLM), classical gradient boosting machine, classical linear regression, classical neural network, classical random forest, classical support vector machine (CSVM). Some of the more favored model structures for quantum simulators include: quantum Boltzmann machine, quantum restricted Boltzmann machine (QRBM), quantum neural network, and quantum support vector
30 machine (QSVM).

Associated models can suffer during training cycles from overfitting or underfitting. In statistics, overfitting is the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably. An overfitted model is a statistical model that contains
5 more parameters than can be justified by the data. The essence of overfitting is to have unknowingly extracted some of the residual variation (e.g., the noise) as if that variation represented underlying model structure.

Underfitting occurs when a statistical model cannot adequately capture the underlying
10 structure of the data. An underfitted model is a model where some parameters or terms that would appear in a correctly specified model are missing. Underfitting would occur, for example, when fitting a linear model to non-linear data. Such a model will tend to have poor predictive performance.

15 Ensemble simulators (e.g., ensemble learners) combine the use of multiple AI/ML simulation models to decrease variance, bias, overfitting, and underfitting. Ensemble simulators use multiple, different model structures to create different associated models that are run in series or parallel. The predictions from multiple associated models can be evaluated using a characterization step and/or combined into a weighted prediction.
20 Ensemble simulators, though more complex in design, are better able to adapt to different types of data and also exploit the strengths and weaknesses of different model structures that are the basis of the associated models used to create winning predictions.

Common types of ensemble simulators include the Bayes optimal classifier, bootstrap
25 aggregating (bagging), boosting, Bayesian parameter averaging, Bayesian model combination, bucket of models, and stacking. An example ensemble simulator is discussed in “*Quantum ensembles of quantum classifiers*” by Maria Schuld and Francesco Petruccione, arXiv:1704.02146v1 [quant-ph] 7 Apr 2017. In this publication, the ensemble simulator employs a collection of quantum model structures.

30

Quantum data, as used to generate associated models based on specific quantum model structures, has the benefit of capturing more permutations than classical data. Quantum data and quantum model structures also enable a stronger ability to detect comparisons between different subsets of data. These same attributes, however, can lead to occasional excessively incorrect predictions if the series of quantum inputs converted from the classical data records are unsuitable for quantum model analysis, such as when the data feed includes many data outliers or is otherwise “noisy”. An ensemble simulator generated from multiple quantum model structures is not immune to this propensity for occasional excessively incorrect predictions as each of the selected simulators uses a quantum model structure and quantum inputs.

Thus the prior art is unable to fully harness the benefits of quantum model structures without succumbing to their inevitable excessively incorrect winning predictions. Greater consistency and confidence in the winning predictions from quantum simulators is therefore needed.

SUMMARY

A general embodiment of the invention is a computer-implemented AI/ML method of using a hybrid simulator with a plurality of selected simulators which include one or more classical simulators and one or more quantum simulators. The hybrid simulator is fed input pairs of classical inputs and quantum inputs, the classical inputs directed to the classical simulators and the quantum inputs directed to the quantum simulators. The classical inputs are pre-processed information extracted from classical data records. The quantum input of each input pair is a reduced quantum representation of the classical input of the input pair. During an iterative series of training cycles and an iterative series of validation cycles, associated models are generated from model structures and the data partitioned for each cycle. Training characterizations and validation characterizations for each associated model are compared with those of other associated models to select the winning models used to generate one or more winning predictions.

A first embodiment of the invention is a computer-implemented AI/ML method of using a hybrid simulator, the method comprising the steps of: (a) accessing a plurality of classical data records from a database; (b) pre-processing the classical data records to create a plurality of input pairs, wherein: (i) each input pair includes a classical input and a quantum input; (ii) each classical input is organized by a plurality of classical data features extracted from the classical data records; and (iii) for each input pair, the quantum input of the input pair is a reduced quantum representation of the classical input of the input pair; (c) partitioning the input pairs for a plurality of cycles including an iterative series of training cycles, an iterative series of validation cycles, and one or more run cycles; (d) initializing the hybrid simulator with a plurality of selected simulators, wherein: (i) the selected simulators include one or more classical simulators and one or more quantum simulators; (ii) each classical simulator is configured to receive only the classical inputs of the input pairs; and (iii) each quantum simulator is configured to receive only the quantum inputs of the input pairs; (e) during each training cycle: (i) configuring the hybrid simulator to define a model structure for each selected simulator; (ii) running the input pairs partitioned for the training cycle through the selected simulators of the hybrid simulator; (iii) for each selected simulator, assembling an associated model from the model structure of the selected simulator and the input pairs partitioned for the training cycle; and (iv) for each selected simulator, assessing a training characterization for the associated model; (f) during each validation cycle: (i) running the input pairs partitioned for the validation cycle through the selected simulators of the hybrid simulator; and (ii) for each selected simulator, assessing a validation characterization for the associated model; (g) calculating a first score for the associated model of each selected simulator by performing a first cross-validation between: (i) the training characterization of the associated model of the selected simulator; and (ii) the validation characterization of the associated model of the selected simulator; (h) calculating a second score for the associated model of each selected simulator by performing a second cross-validation between: (i) the validation characterization of the associated model of the selected simulator; and (ii) the validation characterization of the associated model of each of the selected simulators other than the selected simulator; (i) selecting one or more winning models from the associated models based on the first score and second score of each of

the associated models; and (j) during one or more run cycles, running the input pairs partitioned for the run cycle through each of the one or more winning models of the hybrid simulator to create one or more winning predictions.

- 5 In the first embodiment: (a) the classical data records may comprise at least one of: (i) a plurality of payment history records for a plurality of targeted goods collected from at least one of a plurality of retailers and a plurality of payment systems; and (ii) a plurality of marketing records including information on a plurality of market drivers; (b) the targeted goods may include at least one of: (i) a plurality of fast moving
10 consumer goods; (ii) a plurality of consumer packaged goods; and (iii) a plurality of fast-moving consumer electronics; (c) the market drivers include at least one of product information, pricing information, promotion information, and place information; and (d) the one or more winning predictions include at least one optimized combination of the market drivers.

15

- In the first embodiment, the model structure of the one or more classical simulators includes at least one of: classical Bayesian regression trees, classical Boltzmann machine, classical restricted Boltzmann machine (CRBM), classical general linear model (CGLM), classical gradient boosting machine, classical linear regression,
20 classical neural network, classical random forest, classical support vector machine (CSVM). The model structure of the one or more quantum simulators includes at least one of: quantum Boltzmann machine, quantum restricted Boltzmann machine (QRBM), quantum neural network, and quantum support vector machine (QSVM).

- 25 In the first embodiment: (a) the training characterization for each associated model may be calculated and stored during each training cycle of the iterative series of training cycles to maintain a training history for the associated model; and (b) the training history of each associated model may be monitored for model convergence during the iterative series of training cycles.

30

In the first embodiment: (a) the validation characterization for each associated model may be calculated and stored during each validation cycle of the iterative series of validation cycles to maintain a validation history for the associated model; and (b) the validation history of each associated model may be monitored for model convergence
5 during the iterative series of validation cycles.

In the first embodiment, the training characterization or validation characterization of each associated model may be calculated by at least one of mean squared error, continuous probability distribution matching, stratified k-fold, and Kullback-Leibler
10 divergence (KLD).

In the first embodiment, the method may further comprise the step of creating a weighted prediction, wherein the weighted prediction is a composition of two more winning models weighted according to at least second score of the winning models of
15 two or more winning predictions.

In the first embodiment: (a) the database may be configured to ingest at least one continuous data feed of classical information; and (b) the method may further include the step of evaluating the one or more winning predictions for a given time period with
20 a least one real outcome later observed in the given time period.

A second embodiment of the invention is a data processing system comprising at least one processor configured to perform the method of the first embodiment of the invention.
25

A third embodiment of the invention is a computer program comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method of the first embodiment of the invention.

30 A fourth embodiment of the invention is a computer-readable storage medium comprising instructions which, when executed by a computer, cause the computer to carry out the method the first embodiment of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present disclosure are described herein with reference to the drawings in which:

5

FIG. 1 is block representation of the data processing system including a pre-processor and a hybrid simulator in an embodiment of the invention.

10

FIG. 2 is a flowchart representing the computer-implemented AI/ML method in an embodiment of the invention.

15

FIG. 3 is a table detailing cross-validations made between training characterizations and/or validation characterizations between the associated models of selected simulators in an embodiment of the invention.

20

FIG. 4 is a table detailing training characterizations and validation characterizations of the associated models of a list of sample classical simulators and a sample quantum simulator in an embodiment of the invention.

25

FIG. 5 is a chart graphically representing Kullback-Leibler divergence (KLD) characterizations of the associated models of a sample classical simulator and a quantum simulator versus in an iterative series of training cycles in an embodiment of the invention.

DETAILED DESCRIPTION

In the following detailed description, reference is made to the accompanying drawings, which form a part hereof. The illustrative embodiments described in the detailed description, drawings and claims are not meant to be limiting. Other embodiments can be utilized, and other changes can be made, without departing from the spirit or scope of the subject matter presented herein. Unless specified otherwise, the terms “comprising,” “comprise,” “including” and “include” used herein, and grammatical variants thereof, are intended to represent “open” or “inclusive” language such that they include recited elements but also permit inclusion of additional, un-recited elements. As used herein, a computer may include a single stand-alone computer, a server, multiple dedicated servers, and/or a virtual server running on a larger network of servers and/or cloud-based service. As used herein, a data processing system may reside within a single stand-alone computer, a server, multiple dedicated servers, a cloud-based service, and/or a virtual server running on a larger network of servers. As used herein, a database may store data to and access data from a single stand-alone computer, a data server, multiple dedicated data servers, a cloud-based service, and/or a virtual server running on a larger network of servers.

FIG. 1 is block representation of the data processing system **1-00** including a pre-processor **1-03** and a hybrid simulator **1-04** in an embodiment of the invention. A database **1-02** is fed by a data feed **1-01** of classical data. The database **1-02** stores classical data records **1-07** and, in turn, feeds the classical data records **1-07** into the pre-processor **1-103**. The pre-processor **1-103** performs pre-processing on the classical data records **1-07** to create input pairs. Each input pair includes a classical input CI_n and its paired quantum input QI_n . The quantum input QI_n is prepared from the classical input CI_n of the input pair using a reducer **1-08**, and is a reduced quantum representation of the classical input CI_n of the input pair. The hybrid simulator **1-04** receives each input pair and creates winning predictions **1-106**. A model characterizer **1-105** analyzes activity within the hybrid simulator **1-104** and provides feedback to the hybrid simulator **1-104** such as training characterizations or validation characterizations.

FIG. 2 is a flowchart **2-00** representing the computer-implemented AI/ML method in an embodiment of the invention. The flowchart **2-00** illustrates the steps **2-01** to **2-10** (see below) for implementing an embodiment of the invention.

2-01	access classical data records 1-07 from a database 1-02 with a data feed 1-01 of classical payment history records and classical marketing records
2-02	pre-process the classical data records 1-07 to create input pairs for a hybrid simulator 1-04 , each input pair including a classical input CI_n and a quantum input QI_n , wherein the quantum input QI_n is a reduced quantum representations of the classical inputs CI_n
2-03	partition the input pairs for an iterative series of training cycles, an iterative series of validation cycles, and one or more run cycles
2-04	initialize a hybrid simulator 1-04 with at least one selected classical simulator and at least one selected quantum simulator
2-05	during the iterative series of training cycles, define the associated model of each selected simulator employing a model structure and the input pairs partitioned for the training cycles, assess a training characterization for the associated model
2-06	during the iterative series of validation cycles, run the input pairs partitioned for the validation cycles and assess a validation characterization for the associated model
2-07	perform a first cross-validation between each the training characterization and validation characterization of each associated model to calculate a first score, perform a second cross-validation between validation characterizations of each associated model and the other associated models to calculate a second score
2-08	select one or more winning models based on the first score and second score of each of the associated models
2-09	during the run cycle, run the input pairs partitioned for the run cycle through the winning model(s) to create the winning prediction(s) 1-106
2-10	evaluate the one or more winning predictions 1-106 for a given time period with a least one real outcome later observed in the given time period

FIG. 3 is a table **3-00** detailing cross-validations made between training characterizations and/or validation characterizations between the associated models of selected simulators in an embodiment of the invention. The radial buttons in the first column indicate that: (a) of the available classical simulators (**CS₀**, **CS₁**, **CS₂**), just **CS₁** and **CS₂** are selected; and (b) of the available quantum simulators (**QS₀**, **QS₁**, **QS₂**), just **QS₀** and **QS₂** are selected. The chart **3-00** summarizes that the training characterizations (**CT₁**, **CT₂**, **QT₀**, **QT₂**) of each of the selected simulators are compared to the same selected simulator's validation characterizations (**CV₁**, **CV₂**, **QV₀**, **QV₂**) in a first cross-validation. E.g., in the first cross-validation **CT₁** is compared to **CV₁**, **CT₂** is compared to **CV₂**, **QT₀** is compared to **QV₀**, and **QT₂** is compared to **QV₂**.

The table **3-00** of **FIG. 3** also summarizes that the validation characterizations of each selected simulator of the classical simulators can be compared with the validation characterizations of quantum simulators in a second cross-validation. The table **3-00** summarizes that **CV₁** can be the cross-validated with **QV₀**, and **QV₂**; separately, **CV₂** can also be the cross-validated with **QV₀**, and **QV₂**. Though not summarized in this table **3-00**, the second cross-validation can also be performed between different classical simulators; also the second cross-validation can be performed between the different quantum simulators.

The table **3-00** of **FIG. 3** also summarizes that the training characterizations and validation characterizations of each selected simulator can be compared with a benchmark characterization in a third cross-validation.

FIG. 4 is a table **4-00** detailing training characterizations and validation characterizations of the associated models of selected simulators created from the model structure listed in the table **4-00**. The listed model structure include (from left to right): CGLM, classical Bayesian regression trees, CSVM, CRBM, classical random forest, and QRBM. The measured training characterizations are R-squared and MAPE. The measured validation characterization is MAPE. Of the listed model structures, in this table **4-00** the QRBM model structure performed better than any of the classical

model structures. Also, it should be noted that the training MAPE and the validation MAPE scores (represented as percentage points) are similar for the QRBM, indicating that the associated model created during the training cycle for the QRBM model structure was not significantly underfitted, overfitted, or otherwise compromised. E.g.,
5 the validation characterization during the validation cycle confirmed the viability of the associated model created during the training cycle.

FIG. 5 is a chart **5-00** graphically representing Kullback-Leibler divergence (KLD) characterizations of the associated models of a sample classical simulator and a
10 quantum simulator during an iterative series of training cycles performed by an embodiment of the invention using sample data. This chart **5-00** illustrates that, for these selected simulators and sample data, the quantum simulator's associated model converged in fewer iterations than the classical simulator's associated model.

15 The rapid convergence of the quantum simulator illustrated the chart **5-00** is a common benefit of employing a quantum simulator to create predictions. It should be noted that, partially due to its ability to achieve rapid convergence, the quantum simulator can occasionally converge to a faulty model (for the given data set, type of data, and/or specific prediction output type) and thus produce excessively inaccurate predictions.
20 Hence, the embodiments of the invention create associated models for at least one classical simulator (along with the creation of the one or more quantum simulators) to hedge against the risk of faulty predictions from the quantum simulators. The classical simulator provide alternative associated models that can be characterized; the training characterizations and validation characterizations of the classical simulators are used to
25 evaluate these same characterizations of the quantum simulators. Thus the classical simulator acts to provide a "sanity check" or double check of the interim and final outputs of the quantum simulators during a second cross-validation step. As discussed below, a third cross-validation can also be performed against a benchmark characterization to provide additional assurance against an excessively inaccurate
30 prediction.

A general embodiment of the invention is a computer-implemented AI/ML method of using a hybrid simulator **1-04** with a plurality of selected simulators which include one or more classical simulators and one or more quantum simulators. The hybrid simulator **1-04** is fed input pairs of classical inputs **CI_n** and quantum inputs **QI_n**, the

5 classical inputs **CI_n** directed to the classical simulators and the quantum inputs **QI_n** directed to the quantum simulators. The classical inputs **CI_n** are pre-processed information extracted from classical data records. The quantum input **QI_n** of each input pair is a reduced quantum representation of the classical input of the input pair. During an iterative series of training cycles and an iterative series of validation cycles,

10 associated models are generated from model structures and the data partitioned for each cycle. Training characterizations and validation characterizations for each associated model are compared with those of other associated models to select the winning models used to generate one or more winning predictions.

15 A first embodiment of the invention is a computer-implemented AI/ML method of using a hybrid simulator **1-04**, the method comprising the steps of: (a) accessing a plurality of classical data records **1-07** from a database **1-02**; (b) pre-processing the classical data records **1-07** to create a plurality of input pairs, wherein: (i) each input pair includes a classical input **CI_n** and a quantum input **QI_n**; (ii) each classical input

20 **CI_n** is organized by a plurality of classical data features extracted from the classical data records **1-07**; and (iii) for each input pair, the quantum input **QI_n** of the input pair is a reduced quantum representation of the classical input **CI_n** of the input pair; (c) partitioning the input pairs for a plurality of cycles including an iterative series of training cycles, an iterative series of validation cycles, and one or more run cycles; (d)

25 initializing the hybrid simulator **1-04** with a plurality of selected simulators, wherein: (i) the selected simulators include one or more classical simulators and one or more quantum simulators; (ii) each classical simulator is configured to receive only the classical inputs **CI_n** of the input pairs; and (iii) each quantum simulator is configured to receive only the quantum inputs **QI_n** of the input pairs; (e) during each training cycle:

30 (i) configuring the hybrid simulator **1-04** to define a model structure for each selected simulator; (ii) running the input pairs partitioned for the training cycle through the selected simulators of the hybrid simulator **1-04**; (iii) for each selected simulator,

assembling an associated model from the model structure of the selected simulator and the input pairs partitioned for the training cycle; and (iv) for each selected simulator, assessing a training characterization for the associated model; (f) during each validation cycle: (i) running the input pairs partitioned for the validation cycle through the
5 selected simulators of the hybrid simulator **1-04**; and (ii) for each selected simulator, assessing a validation characterization for the associated model; (g) calculating a first score for the associated model of each selected simulator by performing a first cross-validation between: (i) the training characterization of the associated model of the selected simulator; and (ii) the validation characterization of the associated model of
10 the selected simulator; (h) calculating a second score for the associated model of each selected simulator by performing a second cross-validation between: (i) the validation characterization of the associated model of the selected simulator; and (ii) the validation characterization of the associated model of each of the selected simulators other than the selected simulator; (i) selecting one or more winning models from the
15 associated models based on the first score and second score of each of the associated models; and (j) during one or more run cycles, running the input pairs partitioned for the run cycle through each of the one or more winning models of the hybrid simulator **1-04** to create one or more winning predictions **1-106**.

20 Though, in general, the hybrid simulator **1-04** is designed to feed the same information into both the classical simulator and the quantum simulator, the quantum inputs $\mathbf{QI}_n$ are a reduced quantum representation of the classical input $\mathbf{CI}_n$ of the input pair. This reduction may result in: (a) different size of inputs, different number of inputs, or information content between the classical input $\mathbf{CI}_n$ and the quantum input $\mathbf{QI}_n$ of each
25 input pair; (b) certain classical data records **1-07** may not be fully or even partially represented in the quantum inputs $\mathbf{QI}_n$; and (c) certain data features of the classical data records **1-07** and/or the classical inputs $\mathbf{CI}_n$ may be incompatible with quantum model structures and/or qubit data processing and therefore no inputted to the quantum
30 simulators.

The step of partitioning the input pairs can be performed before, during, or after the input pairs are created during the pre-processing step. As such, the partitioning information can be stored at the database **1-07**, within the pre-processor **1-03**, the hybrid simulator **1-04**, or other memory within the data processing system **1-00**.

5

A library may be maintained with details of various types of classical simulators and quantum simulators, including their model structures and historic characterization or scores in prior simulations. During the step of initializing the hybrid simulator the data processing system **1-00** may automatically select the selected simulators from those available in the library or this step may be performed manually by the user. If any of the selected simulators perform poorly (e.g., exhibit poor training characterizations or validation characterizations), alternative selections from the library may be made. In the next attempt, the hybrid simulator **1-04** is reinitialized with the new batch of selected simulators for new training cycles and validation cycles.

15

Note that **FIG. 1** illustrates a block diagram representation in an embodiment of the invention. The pre-processor **1-03** is representative of the step of pre-processing. The hybrid simulator **1-04** is representative of the steps performed for the training, validation, and run cycles; it can be configured to create associated models, assess and contrast characterizations, select winning models, and select winning predictions. The model characterizer **1-05** can run as part of the hybrid simulator **1-04** or on a separate module or subroutine to assess training characterizations and validation characterizations. Some AI/ML architectures intentionally use two computers, a first computer to run the simulation and a second computer to characterize the interim results of the first computer, in such an architecture the model characterizer **1-05** would run on the second computer to evaluate the interim results of the hybrid simulator **1-04** that runs on the first computer.

The number of training cycles and validation cycles need not be the same. The series of iterative training cycles will typically include more cycles than the series of iterative validation cycles. For instance, the validation cycle could be performed only after a set number of training cycles or after a threshold training characterization is achieved by

30

an associated model. The number of cycles could also vary between the selected simulators, as the associated models for quantum simulators typically converge much sooner than classical simulators and therefore the quantum simulators would often require fewer training and validation cycles.

5

In the first embodiment: (a) the classical data records **1-07** may comprise at least one of: (i) a plurality of payment history records for a plurality of targeted goods collected from at least one of a plurality of retailers and a plurality of payment systems; and (ii) a plurality of marketing records including information on a plurality of market drivers;

10 (b) the targeted goods may include at least one of: (i) a plurality of fast moving consumer goods; (ii) a plurality of consumer packaged goods; and (iii) a plurality of fast-moving consumer electronics; (c) the market drivers include at least one of product information, pricing information, promotion information, and place information; and (d) the one or more winning predictions **1-106** include at least one optimized
15 combination of the market drivers.

In the step of pre-processing of the first embodiment, the classical data records **1-07** may undergo at least one of cleaning, integration, normalization, aggregation, generalization, and reduction.

20

In the first embodiment, the step of pre-processing may include constructing periodic trends for a plurality of variables from the classical data records **1-07**.

In the first embodiment, the classical data features may be extracted employing at least
25 one of a pattern extraction, a capture of nonlinear interactions, and a capture of cross-comparisons.

In the first embodiment, the reduced quantum representation may be a reduced dimensionality of the classical inputs **CI_n**.

30

In the first embodiment, during each of the training cycles, the associated model of each selected simulator may undergo weights initialization and hyper parameter tuning.

In the first embodiment, the model structure of the one or more classical simulators includes at least one of: classical Bayesian regression trees, classical Boltzmann machine, classical restricted Boltzmann machine (CRBM), classical general linear model (CGLM), classical gradient boosting machine, classical linear regression, classical neural network, classical random forest, classical support vector machine (CSVM). The model structure of the one or more quantum simulators includes at least one of: quantum Boltzmann machine, quantum restricted Boltzmann machine (QRBM), quantum neural network, and quantum support vector machine (QSVM).

10

In the first embodiment: (a) the method may further comprise calculating a third score for the associated model of each selected simulator by performing a third cross-validation between a benchmark characterization and at least one of: (i) the training characterization of the associated model of the selected simulator; and (ii) the validation characterization of the associated model of the selected simulator; and (b) the step of selecting the one or more winning models may be also based on the third score of the associated models.

In the first embodiment, each training cycle of the iterative series of training cycles may include a different partitioning of input pairs.

In the first embodiment: (a) the training characterization for each associated model may be calculated and stored during each training cycle of the iterative series of training cycles to maintain a training history for the associated model; and (b) the training history of each associated model may be monitored for model convergence during the iterative series of training cycles.

In the first embodiment: (a) the validation characterization for each associated model may be calculated and stored during each validation cycle of the iterative series of validation cycles to maintain a validation history for the associated model; and (b) the validation history of each associated model may be monitored for model convergence during the iterative series of validation cycles.

In the first embodiment, the training characterization of each associated model may be calculated by at least one of R-squared calculations, mean absolute percentage error (MAPE), mean squared error, continuous probability distribution matching, stratified k-fold, and Kullback-Leibler divergence (KLD).

In the first embodiment, the validating characterization of each associated model may be calculated by at least one of R-squared calculations, mean absolute percentage error (MAPE), mean squared error, continuous probability distribution matching, stratified k-fold, and Kullback-Leibler divergence (KLD).

In the first embodiment, each associated model may be tested during each validation cycle of the iterative series of validation cycles for at least one of: (a) comparison with model fit statistics; and (b) cross validation between the associated models created by the selected simulators.

In the first embodiment, the method may further comprise the step of creating a weighted prediction, wherein the weighted prediction is a composition of two more winning models weighted according to at least second score of the winning models of two or more winning predictions **1-106**.

In the first embodiment: (a) the database **1-02** may be configured to ingest at least one data feed **1-01** of classical information; and (b) the method may further include the step of evaluating the one or more winning predictions **1-106** for a given time period with a least one real outcome later observed in the given time period. The data feed **1-01** of classical information can be an intermittent data feed or a continuous data feed that collects data on an automated fashion as the information becomes available.

A second embodiment of the invention is a data processing system **1-00** comprising at least one processor configured to perform the method of the first embodiment of the invention.

A third embodiment of the invention is a computer program comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method of the first embodiment of the invention.

- 5 A fourth embodiment of the invention is a computer-readable storage medium comprising instructions which, when executed by a computer, cause the computer to carry out the method the first embodiment of the invention.

Source code snippets are included below to assist in the description of the embodiments
10 of the invention. These source code snippets include examples of code drafted to perform steps to: (a) define quantum wave equation; (b) load data; (c) pre-process data; (d) train and validate associated models; (e) measure data divergence; (f) partition a validating set and run the protocol from the start; and (g) validate the performance of each selected simulator's associated model.

15

(a) Define quantum wave equation

def Psi(x):

psi = 0

Gaussian kernel

20 for n in range(N):

psi += np.exp(-np.sum(np.square(x-data[n,:]))/(2*sigma**2))

return psi

def V(x):

v = 0

25 for n in range(N):

v += np.sum(np.square(x-data[n,:]))*np.exp(-np.sum(np.square(x-
data[n,:]))/(2*sigma**2))

v = v/Psi(x)

return v

30

(b) Load data

```
import kvan
# Read data from given columns
cols = [0,1,2,3,4,...]
5 data = kvan.read_Kantar_data(cols)
```

(c) Pre-process data

PRE-PROCESSING

```
# Check number of zero rows
10 len(data)kvan.zero_rows_count(data)

# Drop zero rows to exclude customers that did not purchase anything
data = kvan.drop_zero(data)
len(data)kvan.shuffle(data)
15

# Partition data into training & validating sets
n_training = int(0.8*len(data))training_data = data[0:n_training]
validating_data = data[n_training:-1]
```

20 **(d) Train and validate associated models**

```
# Training
N = len(cols);model_c = kvan.bm(N);
model_c.train(training_data);
model_q = kvan.qbm(N);
25 model_q.train(training_data);

# Validating
kvan.loglikelihood(model_c.prob,validating_data)
kvan.loglikelihood(model_q.prob,validating_data)
30
kvan.loglikelihood(model_c.prob,training_data)
kvan.loglikelihood(model_q.prob,training_data)
```

(e) Measure data divergence

Plot distribution

kvan.plot_dist(Q)

kvan.plot_dist(model_c.prob)

5 kvan.plot_dist(model_q.prob)

KL-divergence

kvan.KL(Q,model_c.prob)

kvan.KL(Q,model_q.prob)

10

(f) Partition a validating set and run the protocol from the start

Validating set

validating_data = kvan.sampling(Q,1000)

15 # Prediction by the machines

predicted_data_c = model_c.predict(1000)

predicted_data_q = model_q.predict(1000)

Comparison via KL divergence

20 kvan.performance(predicted_data_c,validating_data)

kvan.performance(predicted_data_q,validating_data)

Averaged correct prediction by comparing/validating each customer

kvan.averaged_correct_prediction(model_c,validating_data)

25 kvan.averaged_correct_prediction(model_q,validating_data)

(g) Validate the performance of each selected simulator's associated model

kvan.compare(model_c,model_q,Q)

Customize the range of the y-axis

30 kvan.compare(model_c,model_q,Q,[0,0.1])

While various aspects and embodiments have been disclosed herein, it will be apparent that various other modifications and adaptations of the invention will be apparent to the person skilled in the art after reading the foregoing disclosure without departing from the spirit and scope of the invention and it is intended that all such modifications and adaptations come within the scope of the appended claims. The various aspects and
5 embodiments disclosed herein are for purposes of illustration and are not intended to be limiting, with the true scope and spirit of the invention being indicated by the appended claims.

CLAIMS

1. A computer-implemented AI/ML method of using a hybrid simulator, the method comprising the steps of:
 - 5 (a) accessing a plurality of classical data records from a database;
 - (b) pre-processing the classical data records to create a plurality of input pairs, wherein:
 - (i) each input pair includes a classical input and a quantum input;
 - (ii) each classical input is organized by a plurality of classical data features extracted from the classical data records; and
 - 10 (iii) for each input pair, the quantum input of the input pair is a reduced quantum representation of the classical input of the input pair;
 - (c) partitioning the input pairs for a plurality of cycles including an iterative series of training cycles, an iterative series of validation cycles, and one or more run cycles;
 - (d) initializing the hybrid simulator with a plurality of selected simulators, wherein: - 15 (i) the selected simulators include one or more classical simulators and one or more quantum simulators;
 - (ii) each classical simulator is configured to receive only the classical inputs of the input pairs; and
 - (iii) each quantum simulator is configured to receive only the quantum inputs of
 - 20 the input pairs;
 - (e) during each training cycle:
 - (i) configuring the hybrid simulator to define a model structure for each selected simulator;
 - (ii) running the input pairs partitioned for the training cycle through the selected
 - 25 simulators of the hybrid simulator;
 - (iii) for each selected simulator, assembling an associated model from the model structure of the selected simulator and the input pairs partitioned for the training cycle; and
 - (iv) for each selected simulator, assessing a training characterization for the
 - 30 associated model;

- (f) during each validation cycle:
 - (i) running the input pairs partitioned for the validation cycle through the selected simulators of the hybrid simulator; and
 - (ii) for each selected simulator, assessing a validation characterization for the associated model;
 - 5 (g) calculating a first score for the associated model of each selected simulator by performing a first cross-validation between:
 - (i) the training characterization of the associated model of the selected simulator; and
 - 10 (ii) the validation characterization of the associated model of the selected simulator;
 - (h) calculating a second score for the associated model of each selected simulator by performing a second cross-validation between:
 - (i) the validation characterization of the associated model of the selected simulator; and
 - 15 (ii) the validation characterization of the associated model of each of the selected simulators other than the selected simulator;
 - (i) selecting one or more winning models from the associated models based on the first score and second score of each of the associated models; and
 - 20 (j) during one or more run cycles, running the input pairs partitioned for the run cycle through each of the one or more winning models of the hybrid simulator to create one or more winning predictions.
2. The method of any of the preceding claims, wherein:
- 25 (a) the classical data records comprise at least one of:
 - (i) a plurality of payment history records for a plurality of targeted goods collected from at least one of a plurality of retailers and a plurality of payment systems; and
 - (ii) a plurality of marketing records including information on a plurality of
 - 30 market drivers;

- (b) the targeted goods include at least one of:
 - (i) a plurality of fast moving consumer goods;
 - (ii) a plurality of consumer packaged goods; and
 - (iii) a plurality of fast-moving consumer electronics;
 - 5 (c) the market drivers include at least one of product information, pricing information, promotion information, and place information; and
 - (d) the one or more winning predictions include at least one optimized combination of the market drivers.
- 10 3. The method of any of the preceding claims, wherein during the step of pre-processing, the classical data records undergo at least one of cleaning, integration, normalization, aggregation, generalization, and reduction.
4. The method of any of the preceding claims, wherein the step of pre-processing
- 15 includes constructing periodic trends for a plurality of variables from the classical data records.
5. The method of any of the preceding claims, wherein the classical data features are extracted employing at least one of a pattern extraction, a capture of nonlinear
- 20 interactions, and a capture of cross-comparisons.
6. The method of any of the preceding claims, wherein the reduced quantum representation is a reduced dimensionality of the classical inputs.
- 25 7. The method of any of the preceding claims, wherein during each of the training cycles, the associated model of each selected simulator undergoes weights initialization and hyper parameter tuning.

8. The method of any of the preceding claims, wherein:
- (a) the model structure of the one or more classical simulators includes at least one of: classical Bayesian regression trees, classical Boltzmann machine, classical restricted Boltzmann machine (CRBM), classical general linear model (CGLM),
- 5 classical gradient boosting machine, classical linear regression, classical neural network, classical random forest, classical support vector machine (CSVM); and
- (b) the model structure of the one or more quantum simulators includes at least one of: quantum Boltzmann machine, quantum restricted Boltzmann machine (QRBM), quantum neural network, and quantum support vector machine (QSVM).
- 10
9. The method of any of the preceding claims, wherein:
- (a) the method further comprises calculating a third score for the associated model of each selected simulator by performing a third cross-validation between a benchmark characterization and at least one of:
- 15 (i) the training characterization of the associated model of the selected simulator; and
- (ii) the validation characterization of the associated model of the selected simulator; and
- (b) the step of selecting the one or more winning models is also based on the third
- 20 score of the associated models.
10. The method of any of the preceding claims, wherein each training cycle of the iterative series of training cycles includes a different partitioning of input pairs.
- 25 11. The method of any of the preceding claims, wherein:
- (a) the training characterization for each associated model is calculated and stored during each training cycle of the iterative series of training cycles to maintain a training history for the associated model; and
- (b) the training history of each associated model is monitored for model
- 30 convergence during the iterative series of training cycles.

12. The method of any of the preceding claims, wherein:
- (a) the validation characterization for each associated model is calculated and stored during each validation cycle of the iterative series of validation cycles to maintain a validation history for the associated model; and
 - 5 (b) the validation history of each associated model is monitored for model convergence during the iterative series of validation cycles.
13. The method of any of the preceding claims, wherein the training characterization of each associated model is calculated by at least one of R-squared
- 10 calculations, mean absolute percentage error (MAPE), mean squared error, continuous probability distribution matching, stratified k-fold, and Kullback-Leibler divergence (KLD).
14. The method of any of the preceding claims, wherein the validating characterization of each associated model is calculated by at least one of R-squared
- 15 calculations, mean absolute percentage error (MAPE), mean squared error, continuous probability distribution matching, stratified k-fold, and Kullback-Leibler divergence (KLD).
- 20 15. The method of any of the preceding claims, wherein each associated model is tested during each validation cycle of the iterative series of validation cycles for at least one of:
- (a) comparison with model fit statistics; and
 - (b) cross validation between the associated models created by the selected
- 25 simulators.
16. The method of any of the preceding claims, further comprising the step of creating a weighted prediction, wherein the weighted prediction is a composition of two more winning models weighted according to at least second score of the winning
- 30 models of two or more winning predictions.

17. The method of claim 1, wherein:
- (a) the database is configured to ingest at least one continuous data feed of classical information; and
- (b) the method further includes the step of evaluating the one or more winning
- 5 predictions for a given time period with a least one real outcome later observed in the given time period.
18. A data processing system comprising at least one processor configured to perform the method of any of the preceding claims.
- 10
19. A computer program comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method of any one of claims 1-17.
- 15
20. A computer-readable storage medium comprising instructions which, when executed by a computer, cause the computer to carry out the method of any one of claims 1-17.

20