UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/529,142	11/17/2021	ALEXANDER SHEUNG LAI WONG	DARAI.001.C1	1309

95508 7590 09/15/2022

One LLP
23 Corporate Plaza
Suite 150
Newport Beach, CA 92660

EXAMINER	
HOANG, MICHAEL H	

ART UNIT	PAPER NUMBER
2122	

NOTIFICATION DATE	DELIVERY MODE
09/15/2022	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

IPdocketing@onellp.com
PAIRgroup@onellp.com
onellpdocketing@onellp.com

Office Action Summary	Application No. 17/529,142	Applicant(s) WONG et al.	
	Examiner MICHAEL H HOANG	Art Unit 2122	AIA (FITF) Status Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☑ Responsive to communication(s) filed on 17 November 2021.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL.** 2b) ☑ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☑ Claim(s) 1-20 is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6) ☐ Claim(s) _____ is/are allowed.

7) ☑ Claim(s) 1-20 is/are rejected.

8) ☐ Claim(s) _____ is/are objected to.

9) ☐ Claim(s) _____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov.**

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☑ The drawing(s) filed on 17 November 2021 is/are: a)☑ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☑ Notice of References Cited (PTO-892)

2) ☑ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date 11/17/2021.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date _____.

4) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the claims filed 11/17/2021 for application 17/529,142 which is a continuation of application 15/982,478. Claims 1-20 are currently pending.

Notice of Pre-AIA or AIA Status

2. The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

Information Disclosure Statement

3. The information disclosure statement (IDS) submitted on 11/17/2021 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Claim Interpretation

4. The following is a quotation of 35 U.S.C. 112(f):

> (f) Element in Claim for a Combination. – An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

The following is a quotation of pre-AIA 35 U.S.C. 112, sixth paragraph:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

5. The claims in this application are given their broadest reasonable interpretation using the plain meaning of the claim language in light of the specification as it would be understood by one of ordinary skill in the art. The broadest reasonable interpretation of a claim element (also commonly referred to as a claim limitation) is limited by the

description in the specification when 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, is invoked.

As explained in MPEP § 2181, subsection I, claim limitations that meet the following three-prong test will be interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph:

(A) the claim limitation uses the term "means" or "step" or a term used as a substitute for "means" that is a generic placeholder (also called a nonce term or a non-structural term having no specific structural meaning) for performing the claimed function;

(B) the term "means" or "step" or the generic placeholder is modified by functional language, typically, but not always linked by the transition word "for" (e.g., "means for") or another linking word or phrase, such as "configured to" or "so that"; and

(C) the term "means" or "step" or the generic placeholder is not modified by sufficient structure, material, or acts for performing the claimed function.

Use of the word "means" (or "step") in a claim with functional language creates a rebuttable presumption that the claim limitation is to be treated in accordance with 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph. The presumption that the claim limitation is interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, is rebutted when the claim limitation recites sufficient structure, material, or acts to entirely perform the recited function.

Absence of the word "means" (or "step") in a claim creates a rebuttable presumption that the claim limitation is not to be treated in accordance with 35 U.S.C.

112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph. The presumption that the claim limitation is not interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, is rebutted when the claim limitation recites function without reciting sufficient structure, material or acts to entirely perform the recited function.

Claim limitations in this application that use the word "means" (or "step") are being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, except as otherwise indicated in an Office action. Conversely, claim limitations in this application that do not use the word "means" (or "step") are not being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, except as otherwise indicated in an Office action.

6. This application includes one or more claim limitations that do not use the word "means," but are nonetheless being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, because the claim limitations uses a generic placeholder that is coupled with functional language without reciting sufficient structure to perform the recited function and the generic placeholder is not preceded by a structural modifier. Such claim limitations are:

- **a component analyzer module configured to analyze in claim 1.**
- **a component tuner module configured to modify in claim 1.**
- **a machine analyzer configured to analyze in claim 11 and 19.**
- **a machine architecture builder configured to build in claim 11 and 19.**
- **a node analyzer module configured to analyze in claim 19.**
- **an interconnect tuner module configured to modify in claim 19.**

Because these claim limitations are being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, they are being interpreted to cover the corresponding structure described in the specification as performing the claimed function, and equivalents thereof.

If applicant does not intend to have these limitations interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, applicant may: (1) amend the claim limitations to avoid them being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph (e.g., by reciting sufficient structure to perform the claimed function); or (2) present a sufficient showing that the claim limitations recite sufficient structure to perform the claimed function so as to avoid them being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph.

Claim Rejections - 35 USC § 112

7. The following is a quotation of 35 U.S.C. 112(d):

> (d) REFERENCE IN DEPENDENT FORMS.—Subject to subsection (e), a claim in dependent form shall contain a reference to a claim previously set forth and then specify a further limitation of the subject matter claimed. A claim in dependent form shall be construed to incorporate by reference all the limitations of the claim to which it refers.

The following is a quotation of pre-AIA 35 U.S.C. 112, fourth paragraph:

> Subject to the following paragraph [i.e., the fifth paragraph of pre-AIA 35 U.S.C. 112], a claim in dependent form shall contain a reference to a claim previously set forth and then specify a further limitation of the subject matter claimed. A claim in dependent form shall be construed to incorporate by reference all the limitations of the claim to which it refers.

8. Claims 6, 7, and 9 are rejected under 35 U.S.C. 112(d) or pre-AIA 35 U.S.C. 112, 4th paragraph, as being of improper dependent form for failing to further limit the subject matter of the claim upon which it depends, or for failing to include all the limitations of the claim upon which it depends. The component analyzer in claim 6 is interpreted

under 112f, and as such is limited to the description provided in the specification and its equivalents. However, by stating the component analyzer is a node analyzer module, the node analyzer module is not limited to the specification's description of a component analyzer. Therefore, the dependent claim is broader than the independent claim and thus rejected for not further limiting it. The same reasoning above applies to claim 7 as the component tuner module is interpreted under 112(f) and is not further limited by an interconnect tuner module. Applicant may cancel the claims, amend the claims to place the claims in proper dependent form, rewrite the claims in independent form, or present a sufficient showing that the dependent claims complies with the statutory requirements.

9. Claim 9 is rejected for being dependent on a rejected base claim without curing any of the deficiencies.

Claim Rejections - 35 USC § 101

10. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

11. Claims 1-20 are rejected under 35 U.S.C. 101 because the claimed invention is directed to an abstract idea without significantly more.

Regarding **claim 1**,

Step 1 Analysis: Claim 1 is directed to a process, which falls within one of the four statutory categories.

Step 2A Prong 1 Analysis: Claim 1 recites, in part, analyze inputs from the reference learning machine, target learning machine, a set of test signals, and a list of components and modify different components based on the set of output values and component mapping. The limitations of analyzing inputs, a list of components and modifying different components based on the set of output values and component mapping, as drafted, are processes that, under broadest reasonable interpretation, covers performance of the limitation in the mind. If a claim limitation, under its broadest reasonable interpretation, covers performance of the limitation in the mind or pen and paper but for the recitation of generic computer components, then it falls within the "Mental Processes" grouping of abstract ideas. The limitations of:

analyzing inputs, a list of components and modifying different components based on the set of output values and component mapping can be considered to be an evaluation in the human mind.

Accordingly, the claim recites an abstract idea.

Step 2A Prong 2 Analysis: This judicial exception is not integrated into a practical application. In particular, the claim only recites the additional elements – "reference learning machine", "target learning machine", "a set of test signals", "component analyzer module" and "component tuner module". These elements are only generally linked to the judicial exception. The component analyzer module and component tuner module invoke 112f and can be interpreted to be a processor or generic computer as disclosed in [¶0060] of the specification. The – "reference learning machine", "target learning machine", "component analyzer module" and "component tuner module" in the claims are recited at a high-level of generality (i.e. as a generic

processor performing a generic computer function of generating an index) such that it amounts no more than mere instructions to apply the exception using a generic computer component. Accordingly, these additional elements do not integrate the abstract idea into a practical application because they do not impose any meaningful limits on practicing the abstract idea. Please see MPEP§2106.04.(a)(2).III.C. The claim is directed to an abstract idea.

Step 2B Analysis: The claims do not include additional elements that are sufficient to amount to significantly more than the judicial exception. As discussed above with respect to integration of the abstract idea into a practical application, the additional elements of utilizing a reference learning machine, target learning machine, a set of test signals to perform the steps of the claimed process amount to no more than generally linking the elements to the judicial exception. Additionally, utilizing a component analyzer module to analyze inputs and a component tuner module configured to modify different components amount to no more than mere instructions to apply the exception using a generic computer component. Mere instructions to apply an exception using a generic computer component cannot provide an inventive concept. Please see MPEP §2106.05(b). The claim is not patent eligible.

Regarding **claim 2**, the rejection of claim 1 is further incorporated, and further, the claim recites: The system of claim 1, further comprising a feedback loop for feeding back the tuned learning machine as a new target learning machine in an iterative manner. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 3**, the rejection of claim 1 is further incorporated, and further, the claim recites: The system of claim 1, wherein the target learning machine is a new component to be inserted into the reference learning machine. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 4**, the rejection of claim 1 is further incorporated, and further, the claim recites: The system of claim 1, wherein the target learning machine replaces an existing set of components in the reference learning machine. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 5**, the rejection of claim 1 is further incorporated, and further, the claim recites: The system of claim 1, wherein the reference learning machine and

the target learning machine are graph-based learning machines. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 6**, the rejection of claim 5 is further incorporated, and further, the claim recites: The system of claim 5, wherein the component analyzer module is a node analyzer module. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

As noted above, the claim does recite the additional element “node analyzer module”, however it does not amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception, for the reasons set forth in connection with the rejection of claim 1 above. The claim is not patent eligible.

Regarding **claim 7**, the rejection of claim 5 is further incorporated, and further, the claim recites: The system of claim 5, wherein the component tuner module is an interconnect tuner module. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

As noted above, the claim does recite the additional element “interconnect tuner module”, however it does not amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception, for the reasons

set forth in connection with the rejection of claim 1 above. The claim is not patent eligible.

Regarding **claim 8**, the rejection of claim 5 is further incorporated, and further, the claim recites: The system of claim 5, wherein the component mapping is a mapping between nodes from the reference learning machine and nodes from the target learning machine. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 9**, the rejection of claim 7 is further incorporated, and further, the claim recites: The system of claim 7, wherein the interconnect tuner module updates interconnect weights in the target learning machine. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 10**, the rejection of claim 1 is further incorporated, and further, the claim recites: The system of claim 1, wherein the tuned learning machine includes

components updated by the component tuner module. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 1 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 11**,

Step 1 Analysis: Claim 11 is directed to a process, which falls within one of the four statutory categories.

Step 2A Prong 1 Analysis: Claim 11 recites, in part, analyze components based on a set of data to generate a set of machine importance scores. The limitations of analyzing components based on a set of data to generate a set of machine importance scores, as drafted, are processes that, under broadest reasonable interpretation, covers performance of the limitation in the mind but for the recitation of generic computer components. If a claim limitation, under its broadest reasonable interpretation, covers performance of the limitation in the mind or pen and paper, then it falls within the "Mental Processes" grouping of abstract ideas. The limitations of:

analyze components based on a set of data to generate a set of machine importance scores can be considered to be an evaluation in the human mind.

Accordingly, the claim recites an abstract idea.

Step 2A Prong 2 Analysis: This judicial exception is not integrated into a practical application. In particular, the claim recites the additional elements – reciting "initial graph-based learning machine", "machine analyzer", and "machine architecture

builder". The machine analyzer and machine architecture builder invoke 112f and can be interpreted to be a processor or generic computer as disclosed in [¶0060] of the specification. The – "initial graph-based learning machine", "machine analyzer", and "machine architecture builder" in the claims are recited at a high-level of generality (i.e. as a generic processor performing a generic computer function of generating an index) such that it amounts no more than mere instructions to apply the exception using a generic computer component. Accordingly, these additional elements do not integrate the abstract idea into a practical application because they do not impose any meaningful limits on practicing the abstract idea. Please see MPEP§2106.04.(a)(2).III.C. The claim is directed to an abstract idea.

Furthermore, the claim recites: to build a graph-based learning machine architecture based on the set of machine component importance scores and a set of machine factors. Accordingly, these additional elements do not integrate the abstract idea into a practical application because they do not impose any meaningful limits on practicing the abstract idea. The claim as a whole is directed to an abstract idea.

Step 2B Analysis: The claims do not include additional elements that are sufficient to amount to significantly more than the judicial exception. As discussed above with respect to integration of the abstract idea into a practical application, the additional elements of utilizing a machine analyzer to analyze components based on a set of data and a machine architecture builder to build a graph-based machine amount to no more than mere instructions to apply the exception using a generic computer component. Mere instructions to apply an exception using a generic computer component cannot provide an inventive concept. Please see MPEP §2106.05(b).

Additionally, the limitation of to build a graph-based learning machine architecture based on the set of machine component importance scores and a set of machine factors is well-understood, routine, and conventional, as evidenced by Szeto et al. (¶0230). These limitations therefore remain insignificant extra-solution activity even upon reconsideration, and does not amount to significantly more. Even when considered in combination, these additional elements amount to generally linking the elements to the judicial exception, mere instructions to apply the exception using generic computer components and insignificant extra-solution activity, which cannot provide an inventive concept. The claim is not patent eligible.

Regarding **claim 12**, the rejection of claim 11 is further incorporated, and further, the claim recites: wherein a new graph-based learning machine is built wherein the architecture of the new graph-based learning machine is the same as the graph-based learning machine architecture. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 11 above.

As noted above, the claim does recite the additional element “a new graph-based learning machine”, however it does not amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception, for the reasons set forth in connection with the rejection of claim 1 above. The claim is not patent eligible.

Regarding **claim 13**, the rejection of claim 12 is further incorporated, and further, the claim recites: further comprising a feedback loop for feeding back the new graph-

based learning machine as a new initial graph-based learning machine in an iterative manner. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 11 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 14**, the rejection of claim 11 is further incorporated, and further, the claim recites: wherein the machine analyzer is further configured to: feed each data point in a set of data points from the set of data into the initial graph-based learning machine for a predetermined set of iterations; select groups of nodes and interconnects in the initial graph-based learning machine with each data point in the set of data points to compute an output value corresponding to each machine component in the initial graph-based learning machine; compute an average of a set of computed output values of each machine component for each data point in the set of data points to produce a combined output value of each machine component corresponding to each data point in the set of data points; and compute a machine component importance score for each machine component by averaging final combined output values of each machine component corresponding to all data points in the set of data points from the set of data and dividing the average by a normalization value. The claim recites additional mental steps and mathematical steps in addition to the judicial exceptions identified in the rejection of claim 11, thus recites a judicial exception

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 15**, the rejection of claim 11 is further incorporated, and further, the claim recites: wherein the machine analyzer is further configured to: feed each data point in a set of data points from the set of data into the initial graph-based learning machine for a predetermined set of iterations; randomly select groups of nodes and interconnects in the initial graph-based learning machine with each data point in the set of data points to compute an output value corresponding to one of the nodes of the initial graph-based learning machine; average a set of computed output values for each data point in the set of data points to produce a final combined output value corresponding to each data point; and compute a full machine score for each component in the initial graph-based learning machine by averaging a final combined output value corresponding to all data points in the set of data. The claim recites additional mental steps and mathematical steps in addition to the judicial exceptions identified in the rejection of claim 11, thus recites a judicial exception

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 16**, the rejection of claim 15 is further incorporated, and further, the claim recites: The system of claim 15, wherein the machine analyzer is further

configured to: feed each data point in the set of data points from the set of data into a reduced graph- based learning machine, with at least some machine components excluded, for a predetermined set of iterations; randomly select groups of nodes and interconnects in the reduced graph-based learning machine with each data point in the set of data points to compute an output value corresponding to one of the nodes of the reduced graph-based learning machine; compute an average of a set of computed output values for each data point to produce a final combined output value corresponding to each data point; and compute a reduced machine score for each component in the reduced graph-based learning machine by averaging a final combined output value corresponding to all data points in the set of data points in the set of data. The claim recites additional mental steps and mathematical steps in addition to the judicial exceptions identified in the rejection of claim 11, thus recites a judicial exception

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 17**, the rejection of claim 11 is further incorporated, and further, the claim recites: The system of claim 11, wherein the machine architecture builder is further configured to control the size of the new graph-based learning machine architecture based on the set of machine component importance scores and the set of machine factors. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 11 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 18**, the rejection of claim 17 is further incorporated, and further, the claim recites: The system of claim 17, wherein the machine architecture builder controls the size of the new graph-based learning machine architecture by determining whether each node will exist in the new graph-based learning machine architecture. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 11 above.

The claim does not include any additional elements that amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception. The claim is not patent eligible.

Regarding **claim 19**,

Step 1 Analysis: Claim 19 is directed to a process, which falls within one of the four statutory categories.

Step 2A Prong 1 Analysis: Claim 19 recites, in part, analyze inputs from the reference learning machine, the target learning machine, a set of test signals, and a list of nodes, modify different components, analyzing components based on a set of data to generate a set of machine importance scores. The limitations analyzing inputs from the reference learning machine, the target learning machine, a set of test signals, and a list of nodes, modify different components, analyzing components based on a set of data to

generate a set of machine importance scores, as drafted, are processes that, under broadest reasonable interpretation, covers performance of the limitation in the mind but for the recitation of generic computer components. If a claim limitation, under its broadest reasonable interpretation, covers performance of the limitation in the mind or pen and paper, then it falls within the "Mental Processes" grouping of abstract ideas. The limitations of:

analyze inputs from the reference learning machine, the target learning machine, a set of test signals, and a list of nodes, modify different components, analyzing components based on a set of data to generate a set of machine importance scores can be considered to be an evaluation in the human mind.

Accordingly, the claim recites an abstract idea.

Step 2A Prong 2 Analysis: This judicial exception is not integrated into a practical application. In particular, the claim only recites the additional elements – reciting "reference learning machine", "target learning machine", "node analyzer module", "interconnect tuner module", "initial graph-based learning machine", "machine analyzer", and "machine architecture builder". The node analyzer, interconnect tuner module, machine analyzer and machine architecture builder invoke 112f and can be interpreted to be a processor or generic computer as disclosed in [¶0060] of the specification. The – "reference learning machine", "target learning machine", "node analyzer module", "interconnect tuner module", "initial graph-based learning machine", "machine analyzer", and "machine architecture builder" in the claims are recited at a high-level of generality (i.e. as a generic processor performing a generic computer function of generating an index) such that it amounts no more than mere instructions to

apply the exception using a generic computer component. Accordingly, these additional elements do not integrate the abstract idea into a practical application because they do not impose any meaningful limits on practicing the abstract idea. Please see MPEP§2106.04.(a)(2).III.C. The claim is directed to an abstract idea.

Furthermore, the claim recites: to build a graph-based learning machine architecture based on the set of machine component importance scores and a set of machine factors. Accordingly, these additional elements do not integrate the abstract idea into a practical application because they do not impose any meaningful limits on practicing the abstract idea. The claim as a whole is directed to an abstract idea.

Step 2B Analysis: The claims do not include additional elements that are sufficient to amount to significantly more than the judicial exception. As discussed above with respect to integration of the abstract idea into a practical application, the additional elements of utilizing a node analyzer to analyze inputs, an interconnect tuner module to modify different components, a machine analyzer to analyze components based on a set of data and a machine architecture builder to build a graph-based machine amount to no more than mere instructions to apply the exception using a generic computer component. Mere instructions to apply an exception using a generic computer component cannot provide an inventive concept. Please see MPEP §2106.05(b). Additionally, the limitation of to build a graph-based learning machine architecture based on the set of machine component importance scores and a set of machine factors is well-understood, routine, and conventional, as evidenced by Szeto et al. (¶0230). These limitations therefore remain insignificant extra-solution activity even upon reconsideration, and does not amount to significantly more. Even when

considered in combination, these additional elements amount to generally linking the elements to the judicial exception, mere instructions to apply the exception using generic computer components and insignificant extra-solution activity, which cannot provide an inventive concept. The claim is not patent eligible.

Regarding **claim 20**, the rejection of claim 19 is further incorporated, and further, the claim recites: The system of claim 19, wherein a new graph-based learning machine is built wherein the architecture of the new graph-based learning machine is the same as the graph-based learning machine architecture. This limitation amounts to more specifics of the judicial exception identified in the rejection of claim 19 above.

As noted above, the claim does recite the additional element "a new graph-based learning machine", however it does not amount to an integration of the judicial exception into a practical application, nor to significantly more than the judicial exception, for the reasons set forth in connection with the rejection of claim 19 above. The claim is not patent eligible.

Claim Rejections - 35 USC § 102

12. In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

13. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a)(2) the claimed invention was described in a patent issued under section 151, or in an application for patent published or deemed published under section 122(b), in which the patent or application, as the case may be, names another inventor and was effectively filed before the effective filing date of the claimed invention.

14. Claims 1-6, 8, 10-13, 19, and 20 are rejected under 35 U.S.C. 102(a)(2) as being anticipated by Szeto et al. ("US 20170124487 A1, cited by Applicant in the IDS filed on 11/17/2021, hereinafter "Szeto").

Regarding **claim 1**, Szeto teaches A system for building a learning machine, comprising:

a reference learning machine **("specifying a model to be trained by the machine learning platform using the training data, in which the model includes a plurality of algorithms and source code" [¶0047])**;

a target learning machine being built **("generating a new predictive engine variant by training the model to algorithmically arrive upon the label representing the correct answer as provided with the training data based on the plurality of features for each of the multiple transactions" [¶0047])**;

a component analyzer module **([¶0066 discloses use of processors; note: For prior art examination, examiner will interpret elements that invoke 112f to be equivalent to processors.])** configured to analyze inputs from the reference learning machine **("An exemplary system machine learning platform may include, for example, means for: receiving training data as input at the machine learning**

platform, in which the training data includes a multiple transactions, each of the transactions specifying a plurality of features upon which to make a prediction and a label representing a correct answer for the plurality of features according to each respective transaction" [¶0047]), the target learning machine **("versioning the new predictive engine variant based at least on the time the new predictive engine variant was generated a version of the source code utilized within the model and the training data received as input" [¶0047])**, a set of test signals **("For instance, variants of predictive engines and algorithms are evaluated by an evaluator, using one or more metrics with test data. Test data include user queries, predicted results, and actual results or corresponding subsequent user behaviors or sequences of user actions captured and reported to the evaluator." [¶0229])**, and a list of components in the reference learning machine and the target learning machine **("A predictive engine within a PredictionIO or machine learning platform is governed by a set of engine parameters. Engine parameters determine which algorithms are used and what parameters are to be used for each algorithm chosen. In addition, engine parameters dedicate the control of the Data component, Algorithm component, and Serving component of a predictive engine. In other words, engine parameters include parameters for each component controller. As engine parameters essentially teach how an engine is to function, engine parameters are hyperparameters. A given set of engine parameters specifies an engine variant." [¶0262])**, and return a set of output values for each component on the list of components **("Baseline engine variant 527 is of engine type 502, and may take on default engine parameter values stored in a**

PredictionIO or machine learning platform, may be generated manually by an operator, or may be generated automatically. The parameter value, evaluation score, and computation time of each of the engine parameter set and the baseline engine variant are reported at step 530 as output 535." [¶0267]); and

a component tuner module configured to modify different components in the target learning machine based on the set of output values and a component mapping **(FIG. 8 discloses component mapping)**, thereby resulting in a tuned learning machine **("The parameter value, evaluation score, and computation time of each of the engine parameter set and the baseline engine variant are reported at step 530 as output 535. Subsequently, a new predictive engine variant is created at step 540 with the parameter set having the best score. If a baseline engine variant is present, an engine variant is created only if the best score is better than the baseline engine variant's score. The whole engine and its complete parameter set (entire DASE stack, see definitions section), or any sub-component and its associated parameters, may be tuned. This illustrative example shows the tuning of engine parameter sets. In other words, the Data source/data preparator, Algorithm, Serving, and Evaluation components and their parameters can all be tuned in this manner as presented herein." [¶0267; Examiner is interpreting tuning to be equivalent to modifying])**.

Regarding **claim 2**, Szeto discloses The system of claim 1, further comprising a feedback loop for feeding back the tuned learning machine as a new target learning machine in an iterative manner **("More particularly, FIG. 5B illustrates an exemplary**

implementation of the engine parameter tuning process shown in FIG. 5A as a flow diagram 550. The flow diagram 550 starts 552 with step 555, where a list of a given N number of parameter sets is generated to be evaluated. At step 560, an iteration index n is set to 1. Evaluation of the n-th parameter set is carried out at step 565, and the evaluation result is stored in addition to the n-th parameter set itself. If neither a maximum number of tests MAX_N nor timeout has been reached at step 570, the parameter generation and evaluation processes continue through step 572, where the iteration index n is incremented. Otherwise, the presence of a baseline engine variant is considered at step 575. Without a baseline engine variant, the parameter sets and corresponding evaluation results are reported at step 580, and a new engine variant with a parameter set of the best score is created at step 585 before the tuning process terminates at step 590. If a baseline engine variant is present, the evaluation result for the baseline engine variant is evaluated at step 576, reported at step 577, and compared to that of the best score out of the list of parameter sets at step 578. A new engine variant is then created only if the best score is better." [¶0269]).

Regarding **claim 3**, Szeto discloses The system of claim 1, wherein the target learning machine is a new component to be inserted into the reference learning machine **("At block 2325, processing logic deploys the new predictive engine variant into a production environment to replace a prior version of the predictive engine variant." [¶0419])**.

Regarding **claim 4**, Szeto discloses The system of claim 1, wherein the target learning machine replaces an existing set of components in the reference learning machine **("More particularly, FIG. 4 depicts the overall structure of a predictive engine 400, according to one embodiment. Predictive engine 400 may be separated into four major components, Data 420, Algorithm 430, Serving 440, and Evaluator 450, also known as a "DASE" architecture. The first three components Data 420, Algorithm 430, and Serving 440 have been discussed with reference to FIGS. 3A and 3B. This DASE architecture provides a separation of concerns (SoC) that allows developers to exchange and replace individual components in predictive engine design." [¶0253; See ¶[0419] also discloses replacing an existing set of components])**.

Regarding **claim 5**, Szeto discloses The system of claim 1, wherein the reference learning machine **([¶0308 discloses reference learning is graph-based)** and the target learning machine are graph-based learning machines **("More particularly, FIG. 9 depicts an exemplary graph 900 of results, in this case, actual user actions recorded over a given time period, according to one embodiment. In this particular visualization example, user actions are plotted between a starting time 920 at 11:30 pm on a given Monday and an end time 925 at 11:30 pm on the given Monday. Each data point on the plot represents a particular user action or event that occurred after a target prediction has been made in response to a user query with a replay ID. Tracking data 950 are displayed on the graph to show that**

the plotted actual user actions are taken by user 435, after an engine variant 1 has been employed to make predictions." [¶0307]).

Regarding **claim 6**, Szeto discloses The system of claim 5, wherein the component analyzer module is a node analyzer module **("Each data point on the plot represents a particular user action or event that occurred after a target prediction has been made in response to a user query with a replay ID. Tracking data 950 are displayed on the graph to show that the plotted actual user actions are taken by user 435, after an engine variant 1 has been employed to make predictions." [¶0307; Examiner is interpreting a node analyzer to be equivalent to analyzing results from a graph.])**.

Regarding **claim 8**, Szeto discloses The system of claim 5, wherein the component mapping is a mapping between nodes from the reference learning machine and nodes from the target learning machine **("More particularly, FIG. 8 depicts an exemplary diagram 800 showing a PredictionIO or machine learning platform 805 in the process of evaluating and tuning two engine variants, according to one embodiment. Other than user application 880, all components shown in FIG. 8 may be implemented as part of a PredictionIO or machine learning platform 805. A distributed implementation is also possible." [¶0284; See Fig. 8 for component mapping.])**.

Regarding **claim 10**, Szeto discloses The system of claim 1, wherein the tuned learning machine includes components updated by the component tuner module **("The**

whole engine and its complete parameter set (entire DASE stack, see definitions section), or any sub-component and its associated parameters, may be tuned. This illustrative example shows the tuning of engine parameter sets. In other words, the Data source/data preparator, Algorithm, Serving, and Evaluation components and their parameters can all be tuned in this manner as presented herein." [¶0267]).

Regarding **claim 11**, Szeto discloses A system for building a learning machine, comprising:

an initial graph-based learning machine **("In some embodiments, actual results or other types of tracking data may be plotted over shorter or longer time segmentations. In some embodiments, tracking data associated with multiple users, multiple queries, or multiple replay IDs are plotted on the same graph. Moreover, data may be grouped by cohort, session, and other types of data characteristics. The PredictionIO or machine learning platform may automatically detect patterns in tracking data, and cluster them accordingly. On the other hand, operators may specify desired groupings directly. For example, operators can select a specific user and session, to see all the events associated with the user or session." [¶0308])**;

a machine analyzer configured to analyze components of the initial graph-based learning machine based on a set of data to generate a set of machine component importance scores **("In addition, to evaluate the performance of the prediction process to compare different algorithms, algorithm parameter settings, as well as**

different engine variants, an Evaluator component 450 receives data from Serving component 440, and applies one or more metrics to compute evaluation result 455 as an output. An engine variant is a deployable instance of a predictive engine, specified by an engine parameter set. The engine parameter set includes parameters that control each component of a predictive engine. An evaluation metric may quantify prediction accuracy with a numerical score. Evaluation metrics may be pre-defined with default computation steps, or may be customizable by developers who utilize the PredictionIO or machine learning platform." [¶0257; Examiner is interpreting the evaluator component to have an equivalent function of analyzing components of the learning machine.]); and

a machine architecture builder configured to build a graph-based learning machine architecture based on the set of machine component importance scores **("Subsequently, a new predictive engine variant is created at step 540 with the parameter set having the best score. If a baseline engine variant is present, an engine variant is created only if the best score is better than the baseline engine variant's score. The whole engine and its complete parameter set (entire DASE stack, see definitions section), or any sub-component and its associated parameters, may be tuned. This illustrative example shows the tuning of engine parameter sets. In other words, the Data source/data preparator, Algorithm, Serving, and Evaluation components and their parameters can all be tuned in this manner as presented herein. [¶0267])** and a set of machine factors **("Further disclosed are methods and systems for monitoring and replaying queries, predicted results, subsequence end-user actions/behaviors, or actual results, and**

internal tracking information for determining factors that affect the performance of the machine learning system. For example, iterative replay of dynamic queries, corresponding predicted results, and subsequent actual user actions may provide to operators insights into the tuning of data sources, algorithms, algorithm parameters, as well as other system parameters that may affect the performance of the machine learning system. Prediction performances may be evaluated in terms of prediction scores and visualized through plots and diagrams. By segmenting available replay data, prediction performances of different engines or engine variants may be compared and studied conditionally for further engine parameter optimization." [¶0230; Examiner is interpreting the underlined citation to correspond to a set of machine factors]).

Regarding **claim 12**, Szeto discloses The system of claim 11, wherein a new graph-based learning machine is built **("generating a new predictive engine variant by training the model to algorithmically arrive upon the label representing the correct answer as provided with the training data based on the plurality of features for each of the multiple transactions" [¶0047; engine variant corresponds to a new graph-based learning machine])** wherein the architecture of the new graph-based learning machine is the same as the graph-based learning machine architecture **("All engines follow the same DASE architecture described above. Engines are deployed as a web service, which are deployed as a service. [¶0407])**.

Regarding **claim 13**, Szeto discloses The system of claim 12, further comprising a feedback loop for feeding back the new graph-based learning machine as a new initial graph-based learning machine in an iterative manner **("More particularly, FIG. 5B illustrates an exemplary implementation of the engine parameter tuning process shown in FIG. 5A as a flow diagram 550. The flow diagram 550 starts 552 with step 555, where a list of a given N number of parameter sets is generated to be evaluated. At step 560, an iteration index n is set to 1. Evaluation of the n-th parameter set is carried out at step 565, and the evaluation result is stored in addition to the n-th parameter set itself. If neither a maximum number of tests MAX_N nor timeout has been reached at step 570, the parameter generation and evaluation processes continue through step 572, where the iteration index n is incremented. Otherwise, the presence of a baseline engine variant is considered at step 575. Without a baseline engine variant, the parameter sets and corresponding evaluation results are reported at step 580, and a new engine variant with a parameter set of the best score is created at step 585 before the tuning process terminates at step 590. If a baseline engine variant is present, the evaluation result for the baseline engine variant is evaluated at step 576, reported at step 577, and compared to that of the best score out of the list of parameter sets at step 578. A new engine variant is then created only if the best score is better." [¶0269])**.

Regarding **claim 19**, Szeto discloses A system for building a learning machine, comprising:

a reference learning machine **("specifying a model to be trained by the machine learning platform using the training data, in which the model includes a plurality of algorithms and source code" [¶0047])**;

a target learning machine being built **("generating a new predictive engine variant by training the model to algorithmically arrive upon the label representing the correct answer as provided with the training data based on the plurality of features for each of the multiple transactions" [¶0047])**;

a node analyzer module **([¶0066; discloses use of processors])** configured to analyze inputs from the reference learning machine **("An exemplary system machine learning platform may include, for example, means for: receiving training data as input at the machine learning platform, in which the training data includes a multiple transactions, each of the transactions specifying a plurality of features upon which to make a prediction and a label representing a correct answer for the plurality of features according to each respective transaction" [¶0047])**, the target learning machine **("versioning the new predictive engine variant based at least on the time the new predictive engine variant was generated a version of the source code utilized within the model and the training data received as input" [¶0047])**, a set of test signals **("For instance, variants of predictive engines and algorithms are evaluated by an evaluator, using one or more metrics with test data. Test data include user queries, predicted results, and actual results or corresponding subsequent user behaviors or sequences of user actions captured and reported to the evaluator." [¶0229])**, and a list of nodes in the reference learning machine and the target learning machine **("A predictive engine within a PredictionIO or machine**

learning platform is governed by a set of engine parameters. Engine parameters determine which algorithms are used and what parameters are to be used for each algorithm chosen. In addition, engine parameters dedicate the control of the Data component, Algorithm component, and Serving component of a predictive engine. In other words, engine parameters include parameters for each component controller. As engine parameters essentially teach how an engine is to function, engine parameters are hyperparameters. A given set of engine parameters specifies an engine variant." [¶0262]), and return a set of output values for each component on the list of nodes **("Baseline engine variant 527 is of engine type 502, and may take on default engine parameter values stored in a PredictionIO or machine learning platform, may be generated manually by an operator, or may be generated automatically. The parameter value, evaluation score, and computation time of each of the engine parameter set and the baseline engine variant are reported at step 530 as output 535." [¶0267])**;

an interconnect tuner module configured to modify different components in the target learning machine based on the set of output values and a node mapping **(Fig. 8, discloses node mapping)**, thereby resulting in a tuned learning machine **("The parameter value, evaluation score, and computation time of each of the engine parameter set and the baseline engine variant are reported at step 530 as output 535. Subsequently, a new predictive engine variant is created at step 540 with the parameter set having the best score. If a baseline engine variant is present, an engine variant is created only if the best score is better than the baseline engine variant's score. The whole engine and its complete parameter set (entire DASE**

stack, see definitions section), or any sub-component and its associated parameters, may be tuned. This illustrative example shows the tuning of engine parameter sets. In other words, the Data source/data preparator, Algorithm, Serving, and Evaluation components and their parameters can all be tuned in this manner as presented herein." [¶0267; Examiner is interpreting tuning to be equivalent to modifying]);

an initial graph-based learning machine **("In some embodiments, actual results or other types of tracking data may be plotted over shorter or longer time segmentations. In some embodiments, tracking data associated with multiple users, multiple queries, or multiple replay IDs are plotted on the same graph. Moreover, data may be grouped by cohort, session, and other types of data characteristics. The PredictionIO or machine learning platform may automatically detect patterns in tracking data, and cluster them accordingly. On the other hand, operators may specify desired groupings directly. For example, operators can select a specific user and session, to see all the events associated with the user or session." [¶0308])**;

a machine analyzer configured to analyze components of the initial graph-based learning machine based on a set of data to generate a set of machine component importance scores **("In addition, to evaluate the performance of the prediction process to compare different algorithms, algorithm parameter settings, as well as different engine variants, an Evaluator component 450 receives data from Serving component 440, and applies one or more metrics to compute evaluation result 455 as an output. An engine variant is a deployable instance of a predictive**

engine, specified by an engine parameter set. The engine parameter set includes parameters that control each component of a predictive engine. An evaluation metric may quantify prediction accuracy with a numerical score. Evaluation metrics may be pre-defined with default computation steps, or may be customizable by developers who utilize the PredictionIO or machine learning platform." [¶0257; Examiner is interpreting the evaluator component to have an equivalent function of analyzing components of the learning machine.]); and

a machine architecture builder configured to build a graph-based learning machine architecture based on the set of machine component importance scores **("Subsequently, a new predictive engine variant is created at step 540 with the parameter set having the best score. If a baseline engine variant is present, an engine variant is created only if the best score is better than the baseline engine variant's score. The whole engine and its complete parameter set (entire DASE stack, see definitions section), or any sub-component and its associated parameters, may be tuned. This illustrative example shows the tuning of engine parameter sets. In other words, the Data source/data preparator, Algorithm, Serving, and Evaluation components and their parameters can all be tuned in this manner as presented herein. [¶0267])** and a set of machine factors **("Further disclosed are methods and systems for monitoring and replaying queries, predicted results, subsequence end-user actions/behaviors, or actual results, and internal tracking information for determining factors that affect the performance of the machine learning system. For example, iterative replay of dynamic queries, corresponding predicted results, and subsequent actual user actions may**

provide to operators insights into the tuning of data sources, algorithms, algorithm parameters, as well as other system parameters that may affect the performance of the machine learning system. Prediction performances may be evaluated in terms of prediction scores and visualized through plots and diagrams. By segmenting available replay data, prediction performances of different engines or engine variants may be compared and studied conditionally for further engine parameter optimization." [¶0230; Examiner is interpreting the underlined citation to correspond to a set of machine factors]).

Regarding **claim 20**, Szeto discloses The system of claim 19, wherein a new graph-based learning machine is built **("generating a new predictive engine variant by training the model to algorithmically arrive upon the label representing the correct answer as provided with the training data based on the plurality of features for each of the multiple transactions" [¶0047; engine variant corresponds to a new graph-based learning machine])** wherein the architecture of the new graph-based learning machine is the same as the graph-based learning machine architecture **("All engines follow the same DASE architecture described above. Engines are deployed as a web service, which are deployed as a service. [¶0407])**.

Claim Rejections - 35 USC § 103

15. In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any

correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

16. The following is a quotation of 35 U.S.C. 103 which forms the basis for all obviousness rejections set forth in this Office action:

> A patent for a claimed invention may not be obtained, notwithstanding that the claimed invention is not identically disclosed as set forth in section 102, if the differences between the claimed invention and the prior art are such that the claimed invention as a whole would have been obvious before the effective filing date of the claimed invention to a person having ordinary skill in the art to which the claimed invention pertains. Patentability shall not be negated by the manner in which the invention was made.

17. The factual inquiries for establishing a background for determining obviousness under 35 U.S.C. 103 are summarized as follows:

1. Determining the scope and contents of the prior art.

2. Ascertaining the differences between the prior art and the claims at issue.

3. Resolving the level of ordinary skill in the pertinent art.

4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

18. Claims 7, 9, and 15-18 are rejected under 35 U.S.C. 103 as being unpatentable over Szeto in view of Srivastava et al. ("Dropout: A Simple Way to Prevent Neural Networks from Overfitting", cited by applicant in the IDS filed 11/17/2021, hereinafter "Srivastava").

Regarding **claim 7**, Szeto teaches The system of claim 5, however fails to explicitly teach wherein the component tuner module is an interconnect tuner module

Srivastava teaches wherein the component tuner module is an interconnect tuner module **("Model combination nearly always improves the performance of machine learning methods. With large neural networks, however, the obvious idea of averaging the outputs of many separately trained nets is prohibitively expensive. Combining several models is most helpful when the individual models are different from each other and in order to make neural net models different, they should either have different architectures or be trained on different data." [pg. 1930, ¶2; Examiner is interpreting an interconnect tuner module to be for a neural network.])**.

Szeto and Srivastava are both in the same field of endeavor of training a reference model and target model and thus are analogous. Szeto discloses a method of training a model and generating model variants with feedback. Srivastava discloses a deep learning method using dropout learning. It would have been obvious to one of ordinary skill in the art before the effective filing date to substitute the reference model disclosed by Szeto with the neural networks disclosed by Srivastava in order to train a target model with a reference model. One would have been motivated to make this modification since neural networks can learn complicated relationships between their inputs and outputs, thus would be useful for training a target model with a reference model. **[§ Introduction, ¶1, Srivastava]**

Regarding **claim 9**, the combination of Szeto and Srivastava teaches The system of claim 7, where Szeto further teaches wherein the interconnect tuner module updates interconnect weights in the target learning machine **("It is possible that data-**

scientists changed the underlying model itself, resulting in an undesirable behavior, or changed the parameters or changed the selected algorithm, or weightings, or base assumptions, or any number of other possible changes which ultimately result in an undesirable change to the trained model's predictive behaviors and thus necessitate a rollback to a prior version or variant." [¶0217; Examiner is interpreting changing the weightings is equivalent to updating. Additionally, ¶[0267] also discloses updating weights.]).

Regarding **claim 15**, Szeto teaches The system of claim 11, wherein the machine analyzer is further configured to: feed each data point in a set of data points from the set of data into the initial graph-based learning machine for a predetermined set of iterations **("Each data point on the plot represents a particular user action or event that occurred after a target prediction has been made in response to a user query with a replay ID. Tracking data 950 are displayed on the graph to show that the plotted actual user actions are taken by user 435, after an engine variant 1 has been employed to make predictions." [¶0307, see ¶0269 for predetermined set of iterations.])**;

However Szeto fails to explicitly teach randomly select groups of nodes and interconnects in the initial graph-based learning machine with each data point in the set of data points to compute an output value corresponding to one of the nodes of the initial graph-based learning machine;

average a set of computed output values for each data point in the set of data points to produce a final combined output value corresponding to each data point; and

compute a full machine score for each component in the initial graph-based learning machine by averaging a final combined output value corresponding to all data points in the set of data.

Srivastava teaches randomly select groups of nodes and interconnects in the initial graph-based learning machine with each data point in the set of data points to compute an output value corresponding to one of the nodes of the initial graph-based learning machine **("We again use the MNIST data set and do classification by averaging the predictions of k randomly sampled neural networks" [pg. 1947, ¶1, lines 1-2; output value would be equivalent to the predictions.])**;

average a set of computed output values for each data point in the set of data points to produce a final combined output value corresponding to each data point **("An expensive but more correct way of averaging the models is to sample k neural nets using dropout for each test case and average their predictions" [pg. 1946, §7.5 Monte-Carlo Model Averaging vs Weight Scaling, ¶1])**; and

compute a full machine score for each component in the initial graph-based learning machine by averaging a final combined output value corresponding to all data points in the set of data **("As k → ∞, this Monte-Carlo model average gets close to the true model average. It is interesting to see empirically how many samples k are needed to match the performance of the approximate averaging method. By computing the error for different values of k we can see how quickly the error rate of the finite-sample average approaches the error rate of the true model average." [pg. 1946, §7.5 Monte-Carlo Model Averaging vs Weight Scaling, ¶1, Examiner is interpreting the error rate to be equivalent to a full machine score.]).**

Szeto and Srivastava are both in the same field of endeavor of training a reference model and target model and thus are analogous. Szeto discloses a method of training a model and generating model variants with feedback. Srivastava discloses a deep learning method using dropout learning. It would have been obvious to one of ordinary skill in the art before the effective filing date to modify the reference model disclosed by Szeto with the dropout learning method disclosed by Srivastava in order to reduce the size of the neural network by randomly dropping nodes. One would have been motivated to make this modification in order to prevent large networks from overfitting and reducing network size. **[Abstract, Srivastava]**

Regarding **claim 16**, Szeto teaches The system of claim 15, however fails to explicitly teach wherein the machine analyzer is further configured to:

feed each data point in the set of data points from the set of data into a reduced graph- based learning machine, with at least some machine components excluded, for a predetermined set of iterations;

randomly select groups of nodes and interconnects in the reduced graph-based learning machine with each data point in the set of data points to compute an output value corresponding to one of the nodes of the reduced graph-based learning machine;

compute an average of a set of computed output values for each data point to produce a final combined output value corresponding to each data point;

and compute a reduced machine score for each component in the reduced graph-based learning machine by averaging a final combined output value corresponding to all data points in the set of data points in the set of data.

Srivastava teaches feed each data point in the set of data points from the set of data into a reduced graph-based learning machine, with at least some machine components excluded, for a predetermined set of iterations **("Dropout neural networks can be trained using stochastic gradient descent in a manner similar to standard neural nets. The only difference is that for each training case in a mini-batch, we sample a thinned network by dropping out units. Forward and backpropagation for that training case are done only on this thinned network" [pg. 1934, 5.1 Backpropagation, ¶1, Dropout drops units from a neural network, this would be equivalent to a reduced graph-based learning machine.])**;

randomly select groups of nodes and interconnects in the reduced graph-based learning machine with each data point in the set of data points to compute an output value corresponding to one of the nodes of the reduced graph-based learning machine **("The term "dropout" refers to dropping out units (hidden and visible) in a neural network. By dropping a unit out, we mean temporarily removing it from the network, along with all its incoming and outgoing connections, as shown in Figure 1. The choice of which units to drop is random. In the simplest case, each unit is retained with a fixed probability p independent of other units, where p can be chosen using a validation set or can simply be set at 0.5, which seems to be close to optimal for a wide range of networks and tasks. For the input units, however, the optimal probability of retention is usually closer to 1 than to 0.5." [pg. 1930, bottom para])**;

compute an average of a set of computed output values for each data point to produce a final combined output value corresponding to each data point **("An**

expensive but more correct way of averaging the models is to sample k neural nets using dropout for each test case and <u>average their predictions</u>" [pg. 1946, §7.5 Monte-Carlo Model Averaging vs Weight Scaling, ¶1]);

and compute a reduced machine score for each component in the reduced graph-based learning machine by averaging a final combined output value corresponding to all data points in the set of data points in the set of data **("The idea is to use a single neural net at test time without dropout. The weights of this network are scaled-down versions of the trained weights. If a unit is retained with probability p during training, the outgoing weights of that unit are multiplied by p at test time as shown in Figure 2. This ensures that for any hidden unit the expected output (under the distribution used to drop units at training time) is the same as the actual output at test time. By doing this scaling, 2^n networks with shared weights can be combined into a single neural network to be used at test time. <u>We found that training a network with dropout and using this approximate averaging method at test time leads to significantly lower generalization error on a wide variety of classification problems compared to training with other regularization methods</u>" [pg. 1931, ¶2; Examiner is interpreting a lower error would correspond to a reduced machine score.])**.

Szeto and Srivastava are both in the same field of endeavor of training a reference model and target model and thus are analogous. Szeto discloses a method of training a model and generating model variants with feedback. Srivastava discloses a deep learning method using dropout learning. It would have been obvious to one of ordinary skill in the art before the effective filing date to modify the reference model

disclosed by Szeto with the dropout learning method disclosed by Srivastava in order to reduce the size of the neural network by randomly dropping nodes. One would have been motivated to make this modification in order to prevent large networks from overfitting and reducing network size. **[Abstract, Srivastava]**

Regarding **claim 17,** Szeto teaches The system of claim 11, based on the set of machine component importance scores **("Subsequently, a new predictive engine variant is created at step 540 with the parameter set having the best score. If a baseline engine variant is present, an engine variant is created only if the best score is better than the baseline engine variant's score. [¶0267])** and the set of machine factors **(For example, iterative replay of dynamic queries, corresponding predicted results, and subsequent actual user actions may provide to operators insights into the tuning of data sources, algorithms, algorithm parameters, as well as other system parameters that may affect the performance of the machine learning system. Prediction performances may be evaluated in terms of prediction scores and visualized through plots and diagrams. By segmenting available replay data, prediction performances of different engines or engine variants may be compared and studied conditionally for further engine parameter optimization." [¶0230; Examiner is interpreting the underlined citation to correspond to a set of machine factors])**.

However Szeto fails to explicitly teach wherein the machine architecture builder is further configured to control the size of the new graph-based learning machine architecture

Srivastava teaches wherein the machine architecture builder is further configured to control the size of the new graph-based learning machine architecture **("However, we see that dropout improves significantly upon the performance of standard neural nets and outperforms all other methods. The challenge in this data set is to prevent overfitting since the size of the training set is small. One way to prevent overfitting is to reduce the input dimensionality using PCA. Thereafter, standard techniques such as SVMs or logistic regression can be used. However, with dropout we were able to prevent overfitting without the need to do dimensionality reduction. The dropout nets are very large (1000s of hidden units) compared to a few tens of units in the Bayesian network. This shows that dropout has a strong regularizing effect." [pg. 1942, ¶2; Srivastava's dropout method would be equivalent to controlling the size of a graph-based learning machine architecture as it drops units within the network, therefore "controlling" the size of the network.])**

Szeto and Srivastava are both in the same field of endeavor of training a reference model and target model and thus are analogous. Szeto discloses a method of training a model and generating model variants with feedback. Srivastava discloses a deep learning method using dropout learning. It would have been obvious to one of ordinary skill in the art before the effective filing date to modify the reference model disclosed by Szeto with the dropout learning method disclosed by Srivastava in order to reduce the size of the neural network by randomly dropping nodes. One would have been motivated to make this modification in order to prevent large networks from overfitting and reducing network size. **[Abstract, Srivastava]**

Regarding **claim 18**, Szeto teaches The system of claim 17, however fails to explicitly teach wherein the machine architecture builder controls the size of the new graph-based learning machine architecture by determining whether each node will exist in the new graph-based learning machine architecture.

Srivastava teaches wherein the machine architecture builder controls the size of the new graph-based learning machine architecture by determining whether each node will exist in the new graph-based learning machine architecture **("The term "dropout" refers to dropping out units (hidden and visible) in a neural network. By dropping a unit out, we mean temporarily removing it from the network, along with all its incoming and outgoing connections, as shown in Figure 1." [pg. 1930, bottom para; See Fig. 1(b) shows which nodes will "exist"])**.

Szeto and Srivastava are both in the same field of endeavor of training a reference model and target model and thus are analogous. Szeto discloses a method of training a model and generating model variants with feedback. Srivastava discloses a deep learning method using dropout learning. It would have been obvious to one of ordinary skill in the art before the effective filing date to modify the reference model disclosed by Szeto with the dropout learning method disclosed by Srivastava in order to reduce the size of the neural network by randomly dropping nodes. One would have been motivated to make this modification in order to prevent large networks from overfitting and reducing network size. **[Abstract, Srivastava]**

19. Claim 14 is rejected under 35 U.S.C. 103 as being unpatentable over Szeto in view of Hara et al. ("Analysis of Dropout Learning Regarded as Ensemble Learning", cited by Applicant in the IDS filed 11/17/2021, hereinafter "Hara").

Regarding **claim 14**, Szeto teaches The system of claim 11, wherein the machine analyzer is further configured to: feed each data point in a set of data points from the set of data into the initial graph-based learning machine for a predetermined set of iterations **("Each data point on the plot represents a particular user action or event that occurred after a target prediction has been made in response to a user query with a replay ID. Tracking data 950 are displayed on the graph to show that the plotted actual user actions are taken by user 435, after an engine variant 1 has been employed to make predictions." [¶0307, see ¶0269 for predetermined set of iterations.])**;

However Szeto fails to explicitly teach

select groups of nodes and interconnects in the initial graph-based learning machine with each data point in the set of data points to compute an output value corresponding to each machine component in the initial graph-based learning machine;

compute an average of a set of computed output values of each machine component for each data point in the set of data points to produce a combined output value of each machine component corresponding to each data point in the set of data points; and

compute a machine component importance score for each machine component by averaging final combined output values of each machine component corresponding

to all data points in the set of data points from the set of data and dividing the average by a normalization value.

Hara teaches select groups of nodes and interconnects in the initial graph-based learning machine with each data point in the set of data points **("**

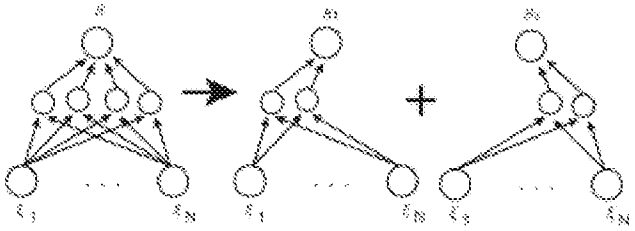

Fig. 3. Network divided into two networks to apply ensemble learning

" [pg. 75, Fig.3]) to compute an output value corresponding to each machine component in the initial graph-based learning machine **("We divide the student (with K hidden units) into K_{en} networks (See Fig. 3. Here, K' = 4 and K_{en} = 2). These divided networks learn the teacher independently, and then we calculate the ensemble output s_{en} by averaging the outputs $s_{k'en}$" [pg. 75, bottom para)**;

compute an average of a set of computed output values of each machine component for each data point in the set of data points to produce a combined output value of each machine component corresponding to each data point in the set of data points ("**These divided networks learn the teacher independently, and then we calculate the ensemble output s_{en} by averaging the outputs $s_{k'en}$" [pg. 75, bottom para])**; and

compute a machine component importance score for each machine component by averaging final combined output values of each machine component corresponding to all data points in the set of data points from the set of data and dividing the average

by a normalization value **("**

$$s_{en} = \frac{1}{K_{en}} \sum_{k'_{en}=1}^{K_{en}} s_{k'_{en}} = \frac{1}{K_{en}} \sum_{k'_{en}=1}^{K_{en}} \sum_{l'=1}^{M/K_{en}} g(y_{k'_{en}l'}). \quad (4)$$

Here, $s_{k'_{en}}$ is the output of a divided network with M/K_{en} hidden units, and $g(y_{k'_{en}l'})$ is the l'th hidden output in the k'_{en}th divided network. Equation (4) corresponds to Eq. (3) when $C_{k'_{en}} = \frac{1}{K_{en}}$ and $K' = \frac{M}{K_{en}}$. **"**

[pg. 75, bottom para; Examiner is interpreting *K*en to be equivalent to dividing the average by a normalization value.]).

Szeto and Hara are both in the same field of endeavor of training a reference model and target model and thus are analogous. Szeto discloses a method of training a model and generating model variants with feedback. Hara discloses a deep learning method using ensemble learning. It would have been obvious to one of ordinary skill in the art before the effective filing date to modify the reference model disclosed by Szeto with the ensemble learning method disclosed by Hara in order to train the model by dividing a neural network to produce a new neural network based off averaging weights of the output and dividing it by a normalized value. One would have been motivated to make this modification in order to avoid overfitting and reduce the network size. **[§ Introduction, pg. 72, Hara]**

Conclusion

20. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Hu et al. ("Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures") discloses network pruning/trimming to remove nodes/neurons

21. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL H HOANG whose telephone number is (571)272-8491. The examiner can normally be reached Mon-Fri 8:30AM-4:30PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To file and manage patent submissions in Patent Center, visit: https://patentcenter.uspto.gov. Visit https://www.uspto.gov/patents/apply/patent-center for more information about Patent Center and https://www.uspto.gov/patents/docx for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197

(toll-free). If you would like assistance from a USPTO Customer Service

Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M.H.H./
Examiner, Art Unit 2122

/KAKALI CHAKI/
Supervisory Patent Examiner, Art Unit 2122