April 2023

A METHOD FOR MITIGATING UNDERFITTING ISSUE IN TIME SERIES MODEL USING REGRESSORS

Pengfei Li
VISA

Pei Yang
VISA

Shuo Wang
VISA

Zhuoyi Wang
VISA

Dan Wang
VISA

TITLE: "A METHOD FOR MITIGATING UNDERFITTING ISSUE IN TIME SERIES MODEL USING REGRESSORS"

VISA

PENGFEI LI
PEI YANG
SHUO WANG
ZHUOYI WANG
DAN WANG

TECHNICAL FIELD

[0001] This disclosure relates generally to the field of computer science. More particularly, the disclosure focuses to mitigate underfitting issue in time series models using regressors.

BACKGROUND

[0002] Generally, when a model inputs dataset in the desired manner/ correctly, it results in a machine learning application performing correctly, and predicting relevant output with good accuracy. There are many instances in which the machine learning applications under perform due to two main reasons such as underfitting and overfitting.

[0003] One of the existing technologies disclose one or more methods such as increasing the model complexity, reducing regularisation and adding features to training data to solve the underfitting issues. The existing technologies defines that a model may be underfitting because it is not complex enough to capture patterns in the data. Further, when the training data is too simple, it may lack the features that will make the model detect the relevant patterns to make accurate predictions. Thus, the method disclosed in the existing technologies describes that by adding features and complexity to data may help overcome underfitting issue.

[0004] Another technology focuses to build time series machine learning or deep learning models based on weekly or daily data. The problem associated with this method is the number of total data points is too small to get the accurate output/ result. For instance, to build a model for forecasting the daily auth volume of an issuer, the model has historical data for about 5 years, and the total number of data points will be about 5*365=1825. However, the model may have only 2000 data points. As a result, the models with many parameters may be built only on few data points due to which the model suffers from underfitting issues. In other words, the parameters of the model are estimated inaccurately due to the presence of very few data points for the training model. For example, if the model has about 500 parameters, but has only 2000 data points, then Data-Per-Parameter (DPP) value will be $\mathrm{DPP} = 2000/500 = 4$. Such DPP value is too low and may potentially cause underfitting issues. Therefore, there is a need for a method that can mitigate underfitting issues.

SUMMARY

According to some non-limiting embodiments, the present disclosure focuses to mitigate underfitting issue in time series models using regressors. The reason for the underfitting may be when the model training is stopped prematurely, it could lead to underfitting, or the data not being trained sufficiently, due to which the model may not be able to capture the vital patterns in data. This would lead to the model not being able to produce satisfactory results. The present disclosure provides efficient way to prevent underfitting issue that may be caused by Low Data-Per-Parameter (DPP) value when a large parameter model is trained with very few data points. The present disclosure replaces the large parameter with a new structure. The new structure may include a few layers of smaller models. Specifically, the “smaller” model may be defined as the number of parameters for the model may be smaller but can handle all the features. Thus, each smaller model is trained with all the data points. The smaller models may generate intermediate predictions which may be fed as the input to the next smaller model present in the next layer. As a result, the DPP value of each model in the structure is substantially higher and hence the underfitting issue may be efficiently resolved.

[0005] In other words, the method of the present disclosure adopts a structure that is composed of a few layers of models and for each of the models there may a limit on how many regressors the model can handle and based on the number of regressor, the number of parameters may be small. However, each model in the structure is trained with all the data points. Due to which the DPP value of each model in the structure may increase and underfitting issue may be solved.

[0006] These and other features and characteristics of the present invention, as well as the methods of operation and functions of the related elements of structures and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only and are not intended as a definition of the limits of the invention. As used in the specification and the claims, the singular form of “a,” “an,” and “the” include plural referents unless the context clearly dictates otherwise.

BRIEF DESCRIPTION OF THE DRAWINGS AND APPENDICES

[0007] Additional advantages and details of non-limiting embodiments are explained in greater detail below with reference to the exemplary embodiments that are illustrated in the accompanying schematic figures, in which:

[0008] FIG. 1A discloses an exemplary representation of intermediate predictions of the models process according to some principles of the present disclosure;

[0009] FIG. 1B discloses an exemplary representation underfitting according to some principles of the present disclosure;

[0010] FIG.2 shows a flowchart that illustrates a method of mitigating underfitting issue in time series models using regressors, in accordance with some embodiments of the present disclosure.

DESCRIPTION OF THE DISCLOSURE

[0011] In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

[0012] While the disclosure is susceptible to various modifications and alternative forms, specific embodiments thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however, that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

[0013] The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

[0014] The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

[0015] The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise.

[0016] As used herein, the term “computing device” may refer to one or more electronic devices that are configured to directly or indirectly communicate with or over one or more networks. A computing device may be a mobile or portable computing device, a desktop computer, a server, and/or the like. Furthermore, the term “computer” may refer to any computing device that includes the necessary components to receive, process, and output data, and normally includes a display, a processor, a memory, an input device, and a network interface. A “computing system” may include one or more computing devices or computers. An “application” or “Application Program Interface” (API) refers to computer code or other data stored on a computer-readable medium that may be executed by a processor to facilitate the interaction between software components, such as a client-side front-end and/or server-side back-end for receiving data from the client. An “interface” refers to a generated display, such as one or more graphical user interfaces (GUIs) with which a user may interact, either directly or indirectly (e.g., through a keyboard, mouse, touchscreen, etc.). Further, multiple computers, e.g., servers, or other computerized devices, such as an autonomous vehicle including a vehicle computing system, directly or indirectly communicating in the network environment may constitute a “system” or a “computing system”.

[0017] It will be apparent that systems and/or methods, described herein, can be implemented in different forms of hardware, software, or a combination of hardware and software. The actual specialized control hardware or software code used to implement these systems and/or methods is not limiting the implementations. Thus, the operation and behavior of the systems and/or methods are described herein without reference to specific software code, it being understood that software and hardware can be designed to implement the systems and/or methods based on the description herein.

[0018] FIG. 1A discloses an exemplary representation of intermediate predictions of the models process according to some principles of the present disclosure.

[0019] FIG. 1A shows an exemplary representation of intermediate predictions of the models. The present disclosure is designed to overcome underfitting issues. The innovative layer-wise structure includes one or more layers of smaller models which acts as an alternative for the large parameter model for time series data. A time series data is the set of measurements taking place in a constant interval of time, here time acts as independent variable and the objective (to study changes in a characteristics) may be a dependent variable. For each of the models there may be a limited number of parameters due to which the Data-Per-Parameter (DPP) may be increased when the one or more models are trained.

[0020] In some embodiments, the machine learning model may not be able to capture the underlying trend of the data due to underfitting issues. The underfitting issues may include, but not limited to, noisy (garbage values) training data, high bias in the model, less training data, when the model is too simple. The underfitting issues may be resolved by processing the data to reduce noise in the data, by adding more number of features in the dataset, by increasing the complexity, reducing the noise in the data, and the like.

For instance, when the model training is stopped prematurely, it could lead to underfitting, or the data not being trained sufficiently, due to which it wouldn’t be able to capture the vital patterns in data. This would lead to the model not being able to produce satisfactory results. The graph below shows underfitting looks visually as shown in **FIG.1B.** From the graph, the dashed line in blue is the model that underfits the data. The black parabola is the line of data points that fits the model well.

[0021] In some embodiments, the present disclosure focuses on state-of-the-art (SOTA) ML/DL models, in which the number of parameters could be very large. For such large-parameter models to be successful, a large amount of training data points is needed. When applying such large-parameter SOTA models on time series data that has only very few data points (for example, only 1825 data points corresponding to about 5 years history), the resulting models might suffer from the underfitting problem which in this context means that the parameters of the model are estimated very inaccurately due to that there are too few data points for training the model. To solve the above-mentioned problem, the present disclosure discloses a structure that stacks a few layers of models and then use the outputs from one or more models as inputs to the final model (which is usually a SOTA model that has good forecasting capability). Thus, the number of parameters for each model in the structure will be relatively small hence the DPP value for each model is substantially larger.

[0022] For instance, M to be data points for a time series forecasting problem, and for ease of understanding say each data point has N features. For example, if we have about 2000 days data for an issuer, then M=2000. Further for each day, the issuer's information like transaction volume (TV), volume-wise percentage of CNP (Card Not Present) transactions ($VPCNP$), count of fraud transactions (CFT), count of active accounts (CAA), are considered, then in total there are N features for each data point. Most of the time, for practical time series problems the build models may be N which may be very large ($N \geq 500$, for example). When a time series model is built, which includes the N features, then DPP may be calculated as $DPP = M/N$ (for example, $DPP = 2000/500 = 4$) which is too small to be considered as the data points. Therefore, a computing system may divide the N raw features into N_1 smaller sets. Particularly, the numbers of features in each smaller set need not be equal. When the numbers of features are not too larger or too smaller than N/N_1 then they can be considered.

[0023] In some embodiments, upon dividing the N raw into smaller sets N_1. The computing system may train a machine learning model with each of the smaller sets. For each smaller set, the number of data points is still M, just the number of features is smaller, so the DPP value is larger and then the underfitting issue is mitigated for each of these models. The each of the models may generate the intermediate output of the time series denoted as $F_{1,j}$ ($j = 1, \cdots, N_1$). The intermediate output contains information from the raw features. However, the total number of intermediate outputs is significantly smaller than the number of raw features ($N_1 < N$) as shown in **FIG.1B.**

[0024] In some embodiments, the computing system may repeat the above-mentioned steps with $F_{1,j}$ ($j = 1, \cdots, N_1$) as inputs in order to reduce the number of inputs to the next layer of models. Thus, each model in the structure of the present disclosure have fewer input features, but the total number of data points for training each model remains the same (M), so the DPP value will be larger and larger for each model ($DPP = M/(number\ of\ input\ features)$). Thereafter, the intermediate output generated by the last layer will be used as inputs to the very last model of the structure. As a result, high-accuracy SOTA model can be chosen as the very last model.

[0025] Therefore, in the present disclosure, the final forecasting result will be more accurate rather than using one large-parameter model to train the model and make the forecasting, because the latter suffers from the underfitting problem due to smaller DPP value. However, the present disclosure has mitigated the underfitting problem by utilizing a few layers of models with each one having higher DPP value due to that each model has fewer parameters because of fewer inputs and the total number of data points for each model remains the same.

[0026] **FIG.2** shows a flowchart that illustrates a method of mitigate underfitting issue in time series models using regressors, in accordance with some embodiments of the present disclosure.

[0027] As illustrated in **FIG. 2**, method **200a** includes one or more blocks illustrating a method for processing the autopayments using account abstraction. The method **200a** may be described in the general context of computer-executable instructions. Generally, computer-executable instructions can include routines, programs, objects, components, data structures, procedures, modules, and functions, which perform functions or implement abstract data types.

[0028] The order in which the method **200a** is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method **200a**. Additionally, individual blocks may be deleted from the methods without departing from the scope of the subject matter described herein. Furthermore, the method **200a** can be implemented in any suitable hardware, software, firmware, or combination thereof.

[0029] At **block 201**, the method **200a** may include dividing the N raw features into N_1 smaller sets. For instance, the numbers of features in each smaller set don't need to be equal. As long as they are not too larger than or too smaller than N/N_1.

[0030] At **block 203**, the method **200a** may include training a machine learning model with each of the smaller sets. In some embodiments, for each smaller set, the number of data points is still M, the number of features is smaller, so the DPP value is larger and then the underfitting issue is mitigated for each of the models.

[0031] At **block 20**5, the method **200a** may include generating an intermediate outputs for the time series and is denoted as $F_{1,j}$ $(j = 1, \cdots, N_1)$. These intermediate predictions contain the information from the raw features.

[0032] At **block 207**, the method **200a** may include reducing the number of inputs to the next layer of models by repeating the above-mentioned steps by considering $F_{1,j}$ $(j = 1, \cdots, N_1)$ as inputs.

[0033] At **block 209**, the method **200a** may include providing the intermediate outputs as the input to the last layer of the structure. As a result, the new structure with few layers of smaller models be defined as the number of parameters for the model may be smaller but can handle all the features. Thus, each smaller models are trained with all the data points. The smaller models may generate intermediate predications which may be fed as the input to the next smaller model present in the next layer. As a result, the DPP value of each model in the structure is substantially higher and hence the underfitting issue may be efficiently resolved. However, the present disclosure is quite different from the neural network structure as it comprises smaller models and dependent on each other.

[0034] The above description is illustrative and is not restrictive. Many variations of the invention may become apparent to those skilled in the art upon review of the disclosure.

[0035] One or more features from any embodiment may be combined with one or more features of any other embodiment without departing from the scope of the invention.

[0036] A recitation of "a", "an" or "the" is intended to mean "one or more" unless specifically indicated to the contrary.

[0037] All patents, patent applications, publications, and descriptions mentioned above are herein incorporated by reference in their entirety for all purposes. None is admitted to being prior art.

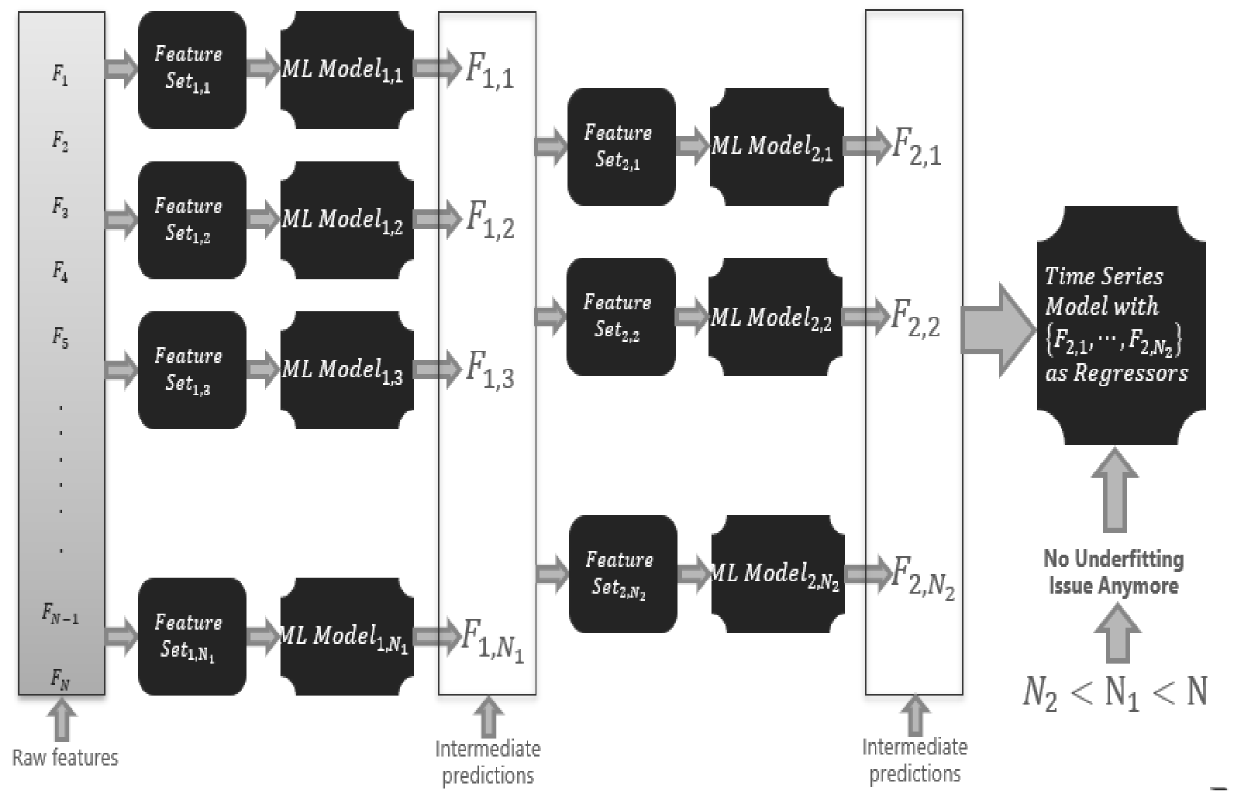
[0038] Although the invention has been described in detail for the purpose of illustration based on what is currently considered to be the most practical and preferred embodiments, it is to be understood that such detail is solely for that purpose and that the invention is not limited to the disclosed embodiments, but, on the contrary, is intended to cover modifications and equivalent arrangements that are within the spirit and scope of the invention. For example, it is to be understood that the present invention contemplates that, to the extent possible, one or more features of any embodiment can be combined with one or more features of any other embodiment.

A METHOD FOR MITIGATING UNDERFITTING ISSUE IN TIME SERIES MODEL USING REGRESSORS

ABSTRACT

The present disclosure focuses on mitigating underfitting issue in time series model using regressors. The present disclosure replaces the large parameter with a new structure. The new structure may include a few layers of smaller models. Specifically, the “smaller” model may be defined as the number of parameters for the model may be smaller but can handle all the features. Thus, each smaller model is trained with all the data points. The smaller models may generate intermediate predictions which may be fed as the input to the next smaller model present in the next layer. As a result, the DPP value of each model in the structure is substantially higher and hence the underfitting issue may be efficiently resolved.

FIG.1



F_1
F_2
F_3
F_4
F_5
F_{N-1}
F_N
Raw features
Feature $Set_{1,1}$
Feature $Set_{1,2}$
Feature $Set_{1,3}$
Feature Set_{1,N_1}
$ML\ Model_{1,1}$
$ML\ Model_{1,2}$
$ML\ Model_{1,3}$
$ML\ Model_{1,N_1}$
$F_{1,1}$
$F_{1,2}$
$F_{1,3}$
F_{1,N_1}
Intermediate predictions
Feature $Set_{2,1}$
Feature $Set_{2,2}$
Feature Set_{2,N_2}
$ML\ Model_{2,1}$
$ML\ Model_{2,2}$
$ML\ Model_{2,N_2}$
$F_{2,1}$
$F_{2,2}$
F_{2,N_2}
Intermediate predictions
Time Series Model with $\{F_{2,1}, \cdots, F_{2,N_2}\}$ as Regressors
No Underfitting Issue Anymore
$N_2 < N_1 < N$

FIG.1A

12

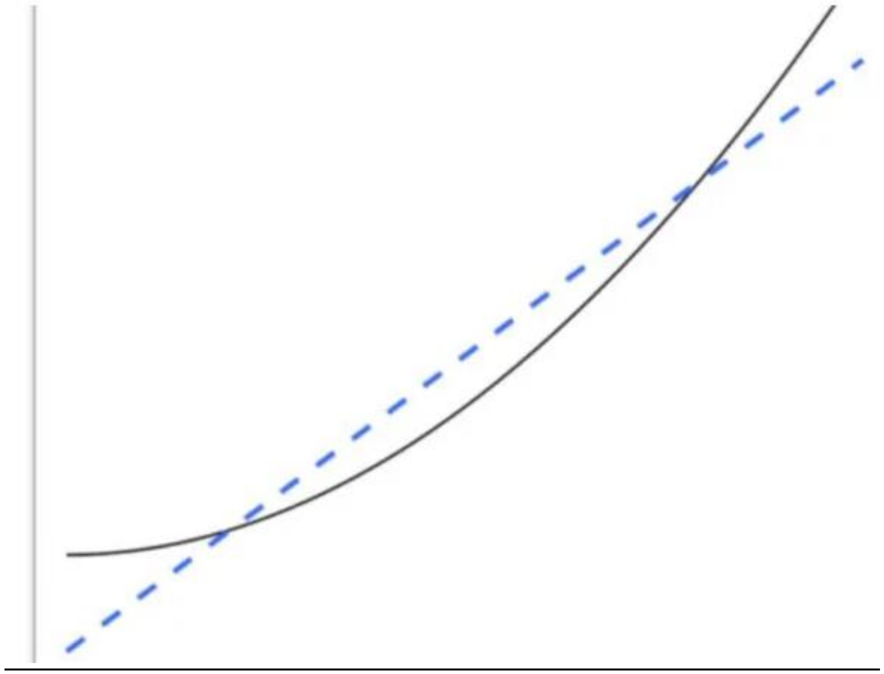


FIG.1B

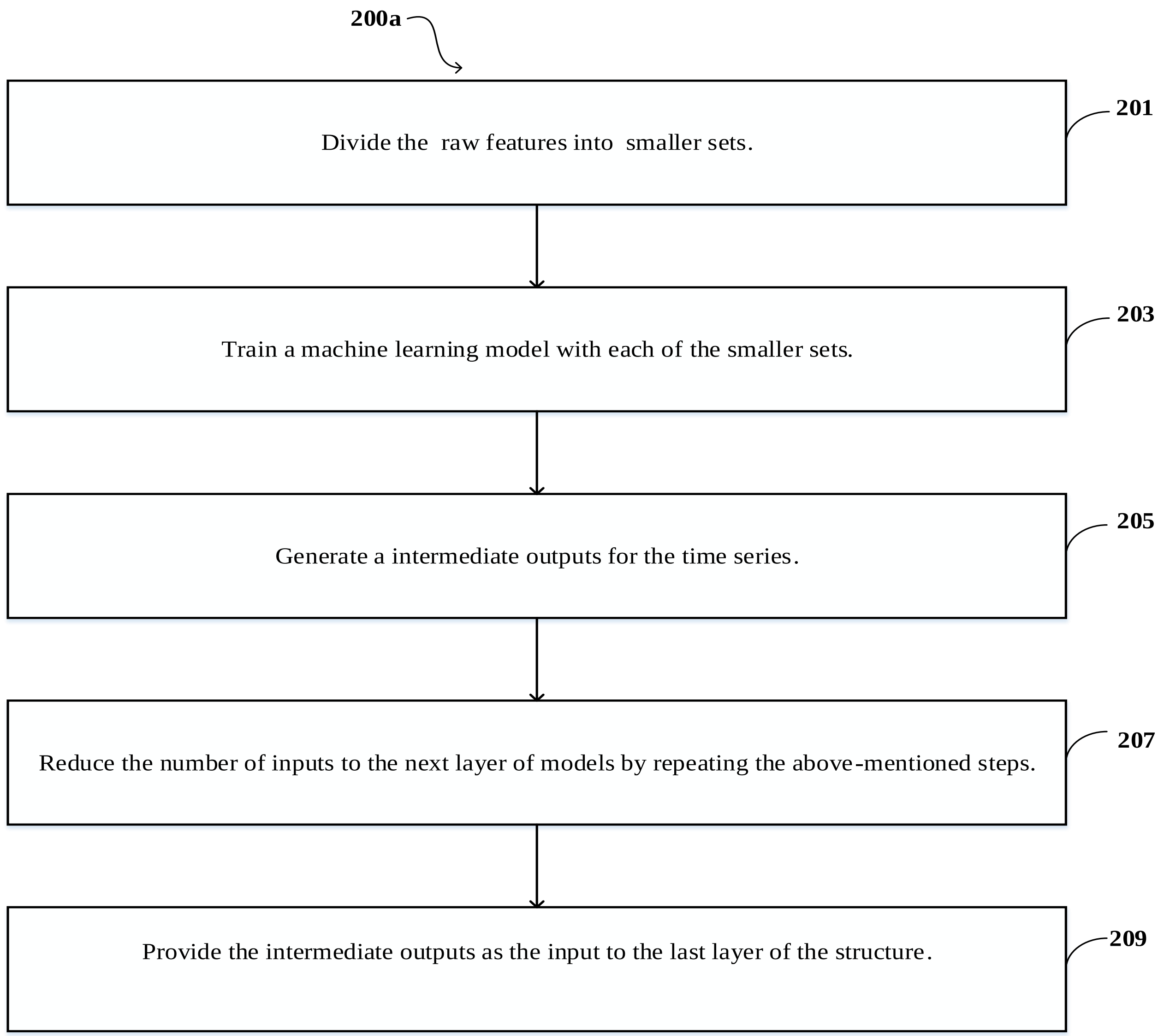

200a
Divide the raw features into smaller sets.
201
Train a machine learning model with each of the smaller sets.
203
Generate a intermediate outputs for the time series.
205
Reduce the number of inputs to the next layer of models by repeating the above-mentioned steps.
207
Provide the intermediate outputs as the input to the last layer of the structure.
209

FIG.2