(12) UK Patent Application (19) GB (11) 2552995 (13) A

(43) Date of A Publication 21.02.2018

(21) Application No: 1614184.8

(22) Date of Filing: 19.08.2016

(71) Applicant(s):
Nokia Technologies Oy
Karaportti 3, 02610 Espoo, Finland

(72) Inventor(s):
Emre Baris Aksu
Francesco Cricri

(74) Agent and/or Address for Service:
Venner Shipley LLP
200 Aldersgate, LONDON, EC1A 4HD,
United Kingdom

(51) INT CL:
H04L 9/00 (2006.01) ***G10L 15/30*** (2013.01)
G10L 17/18 (2013.01)

(56) Documents Cited:
US 20150294670 A1 **US 20120059655 A1**
US 20110285504 A1

(58) Field of Search:
INT CL **G10L, H04L, H04W**
Other: **WPI, EPODOC.**

(54) Title of the Invention: **Learned model data processing**
Abstract Title: **Learned Model Data Processing**

(57) A neural network trained to act as a learned model which *eg.* recognises the speech of a particular user 12 is stored at a first processing system and, upon a request by a user for a network service, is sent to an associated virtual processing environment (*eg.* a Virtual Runtime Environment 20 run by an authenticated broker 14) at a second system for temporary storage and use in generating output which is subsequently provided to a third party provider 18. This allows users who do not wish their personal biometric data (*eg.* voice, image or fingerprint data) to be stored at remote servers to maintain access to service requiring *eg.* voice recognition, since the broker uses training data to create the neural network and then destroys the data.

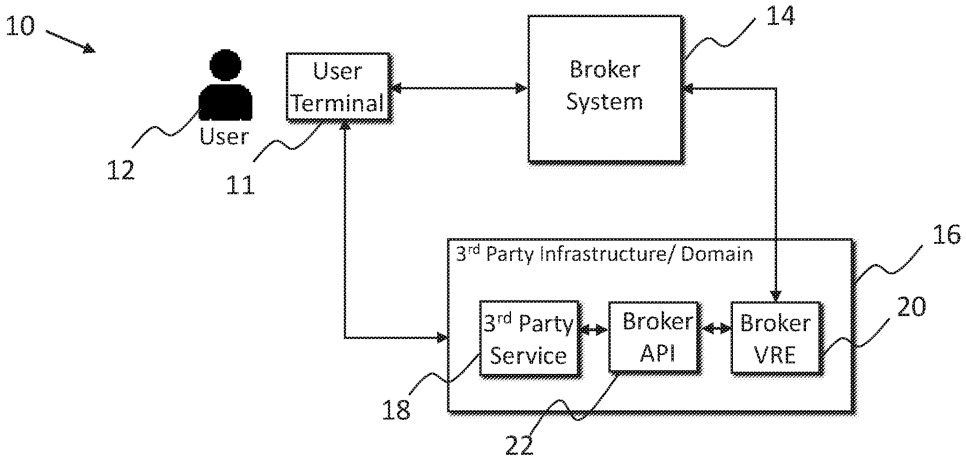

Fig. 3

GB 2552995 A

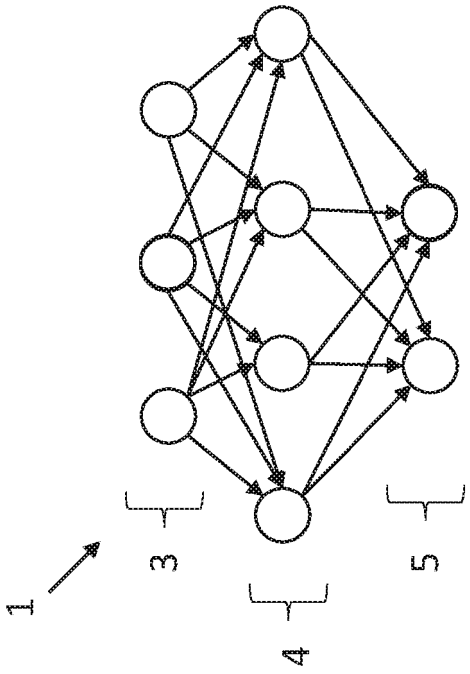

Fig. 1

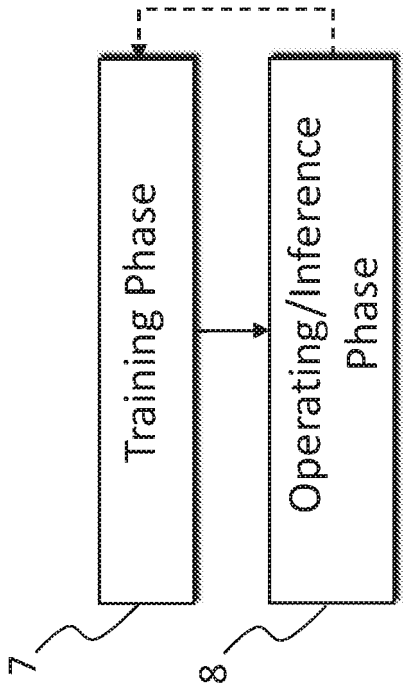

Fig. 2

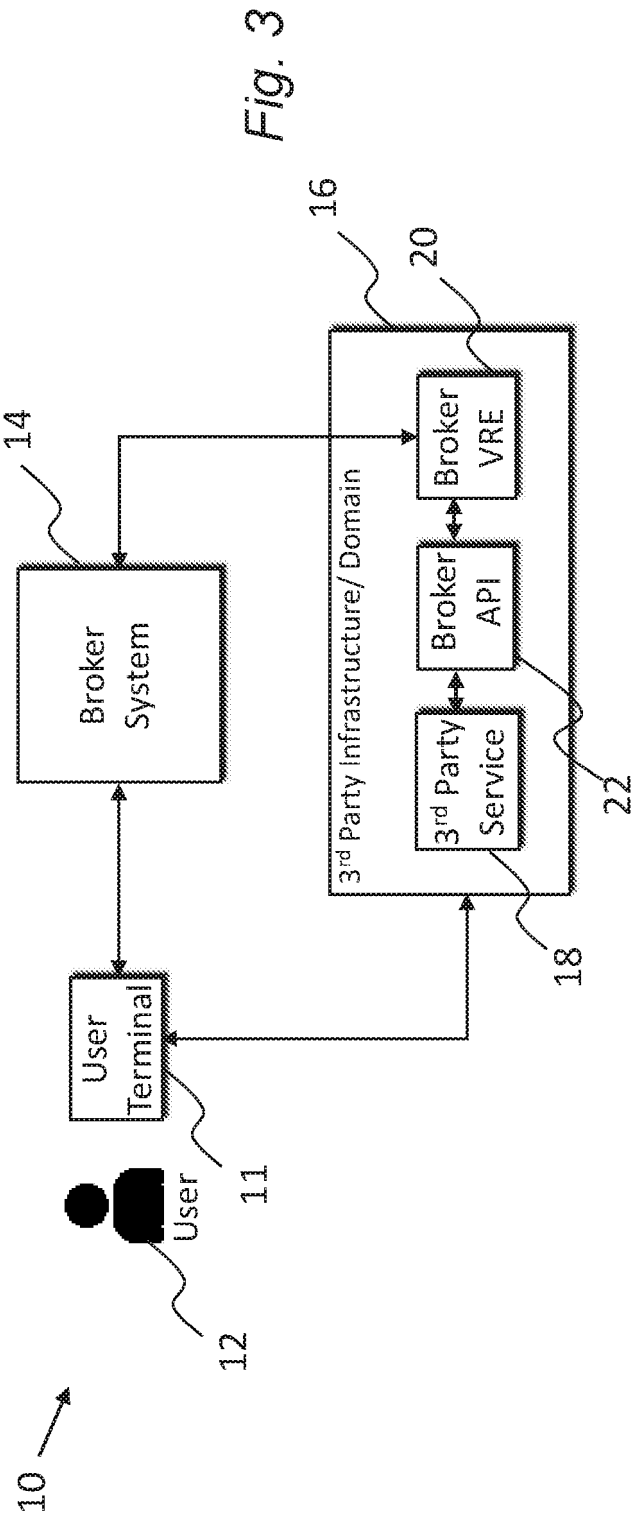
10
User
12
User Terminal
11
Broker System
14
16
3rd Party Infrastructure/ Domain
20
Broker VRE
Broker API
3rd Party Service
18
22

Fig. 3

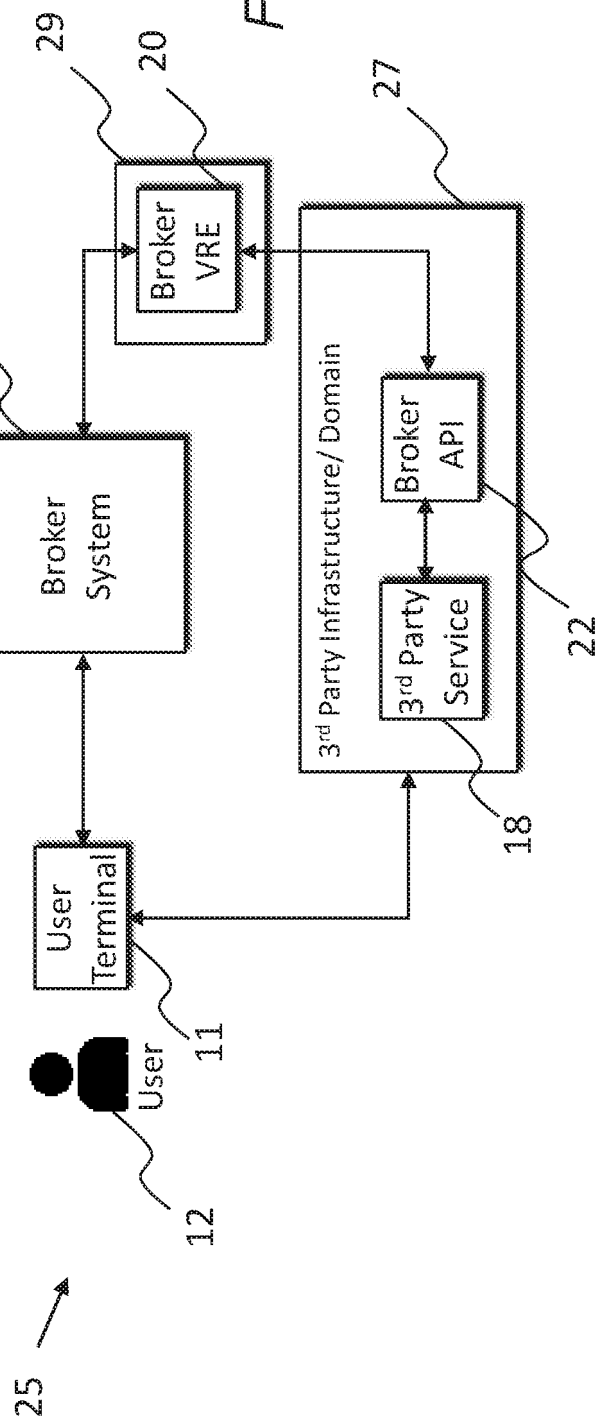
25
User
12
User Terminal
11
Broker System
14
29
20
Broker VRE
27
3rd Party Infrastructure/ Domain
Broker API
3rd Party Service
18
22

Fig. 4

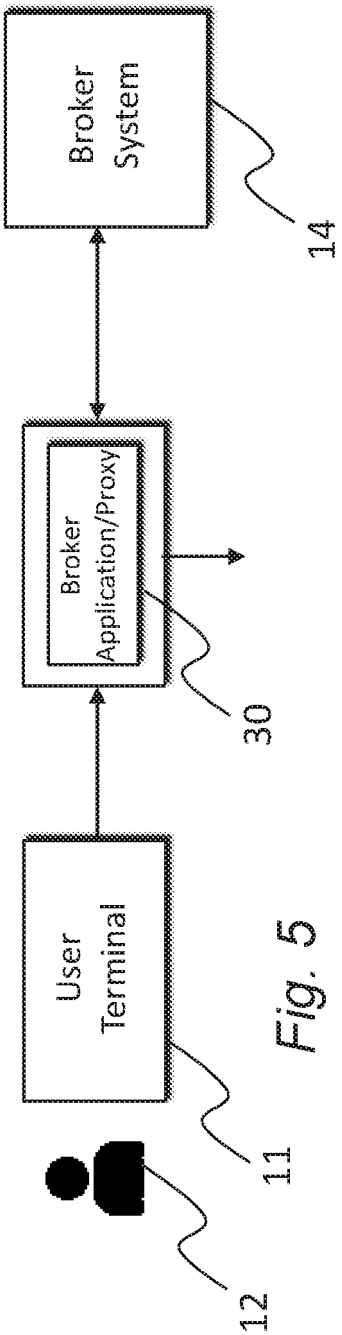
Broker System
14
Broker Application/Proxy
30
User Terminal
11
12

Fig. 5

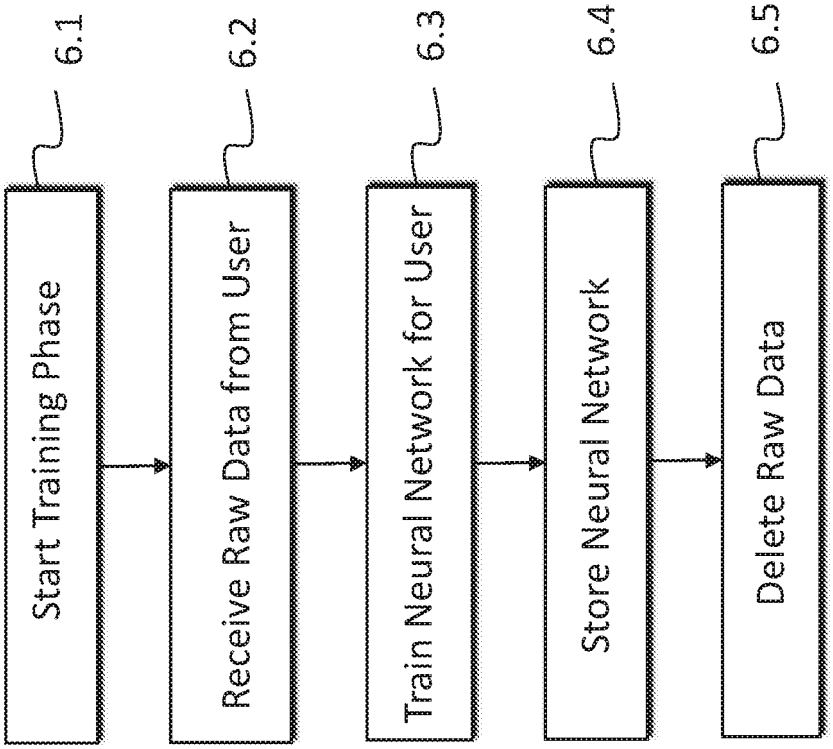
Start Training Phase
6.1
Receive Raw Data from User
6.2
Train Neural Network for User
6.3
Store Neural Network
6.4
Delete Raw Data
6.5

Fig. 6

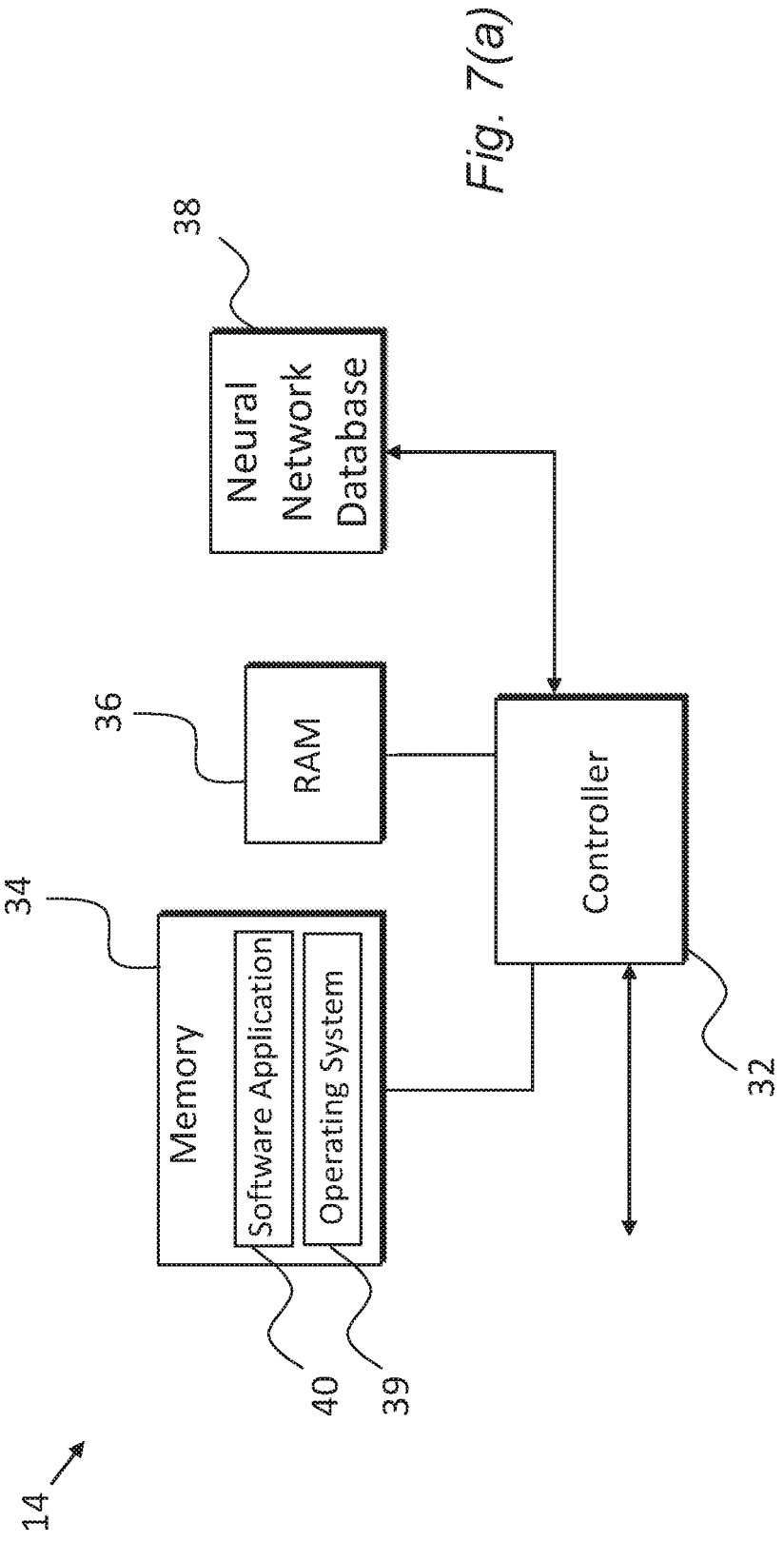

Fig. 7(a)

User	Modality	Task	NN
1	Speech	Voice>Text	{Parameters}
1	Speech	Voice>Song	{Parameters}
2	...	...	...
etc.			

38

Fig. 8

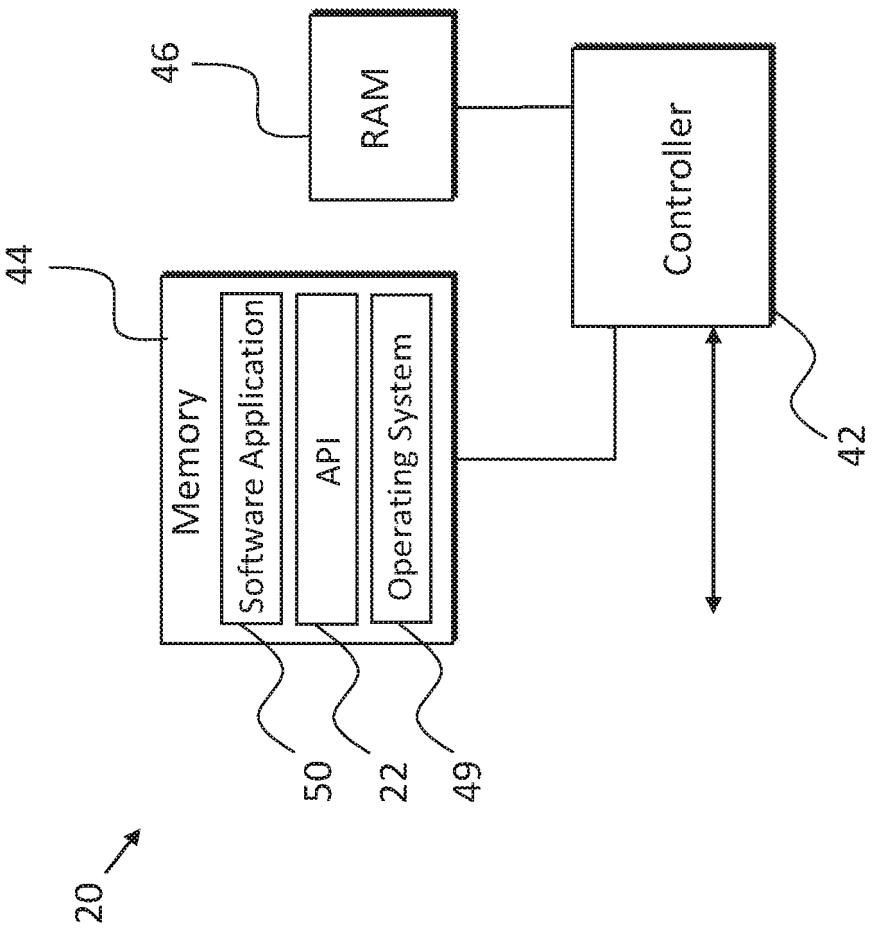
20
44
Memory
Software Application
API
Operating System
50
22
49
46
RAM
Controller
42

Fig. 7(b)

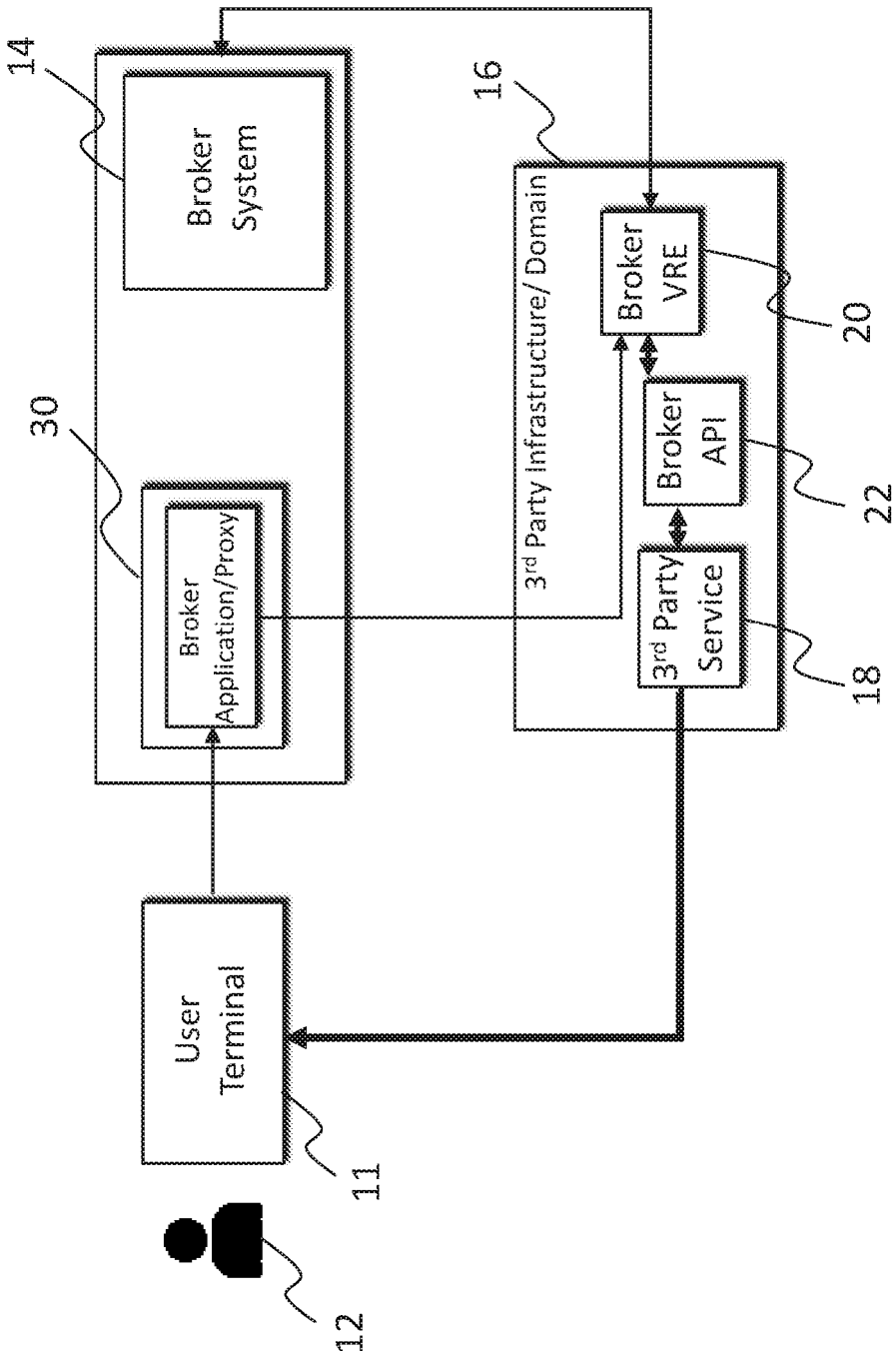
14
Broker System
16
3rd Party Infrastructure/ Domain
Broker VRE
20
30
Broker Application/Proxy
Broker API
22
3rd Party Service
18
User Terminal
11
12

Fig. 9

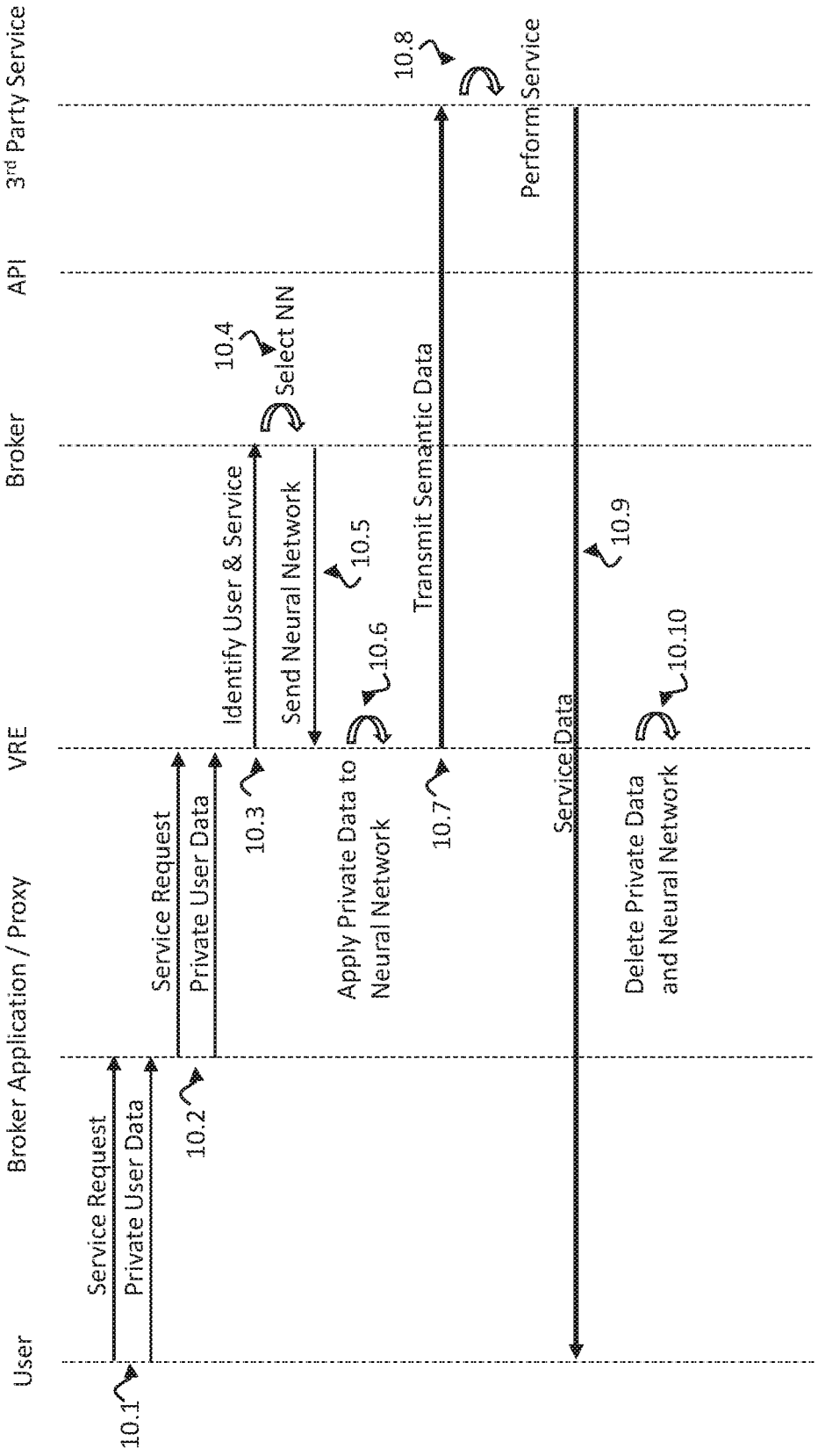
User
Broker Application / Proxy
VRE
Broker
API
3rd Party Service
10.1
Service Request
Private User Data
10.2
Service Request
Private User Data
10.3
Identify User & Service
10.4
Select NN
Send Neural Network
10.5
10.6
Apply Private Data to Neural Network
10.7
Transmit Semantic Data
10.8
Perform Service
Service Data
10.9
Delete Private Data and Neural Network
10.10

Fig. 10

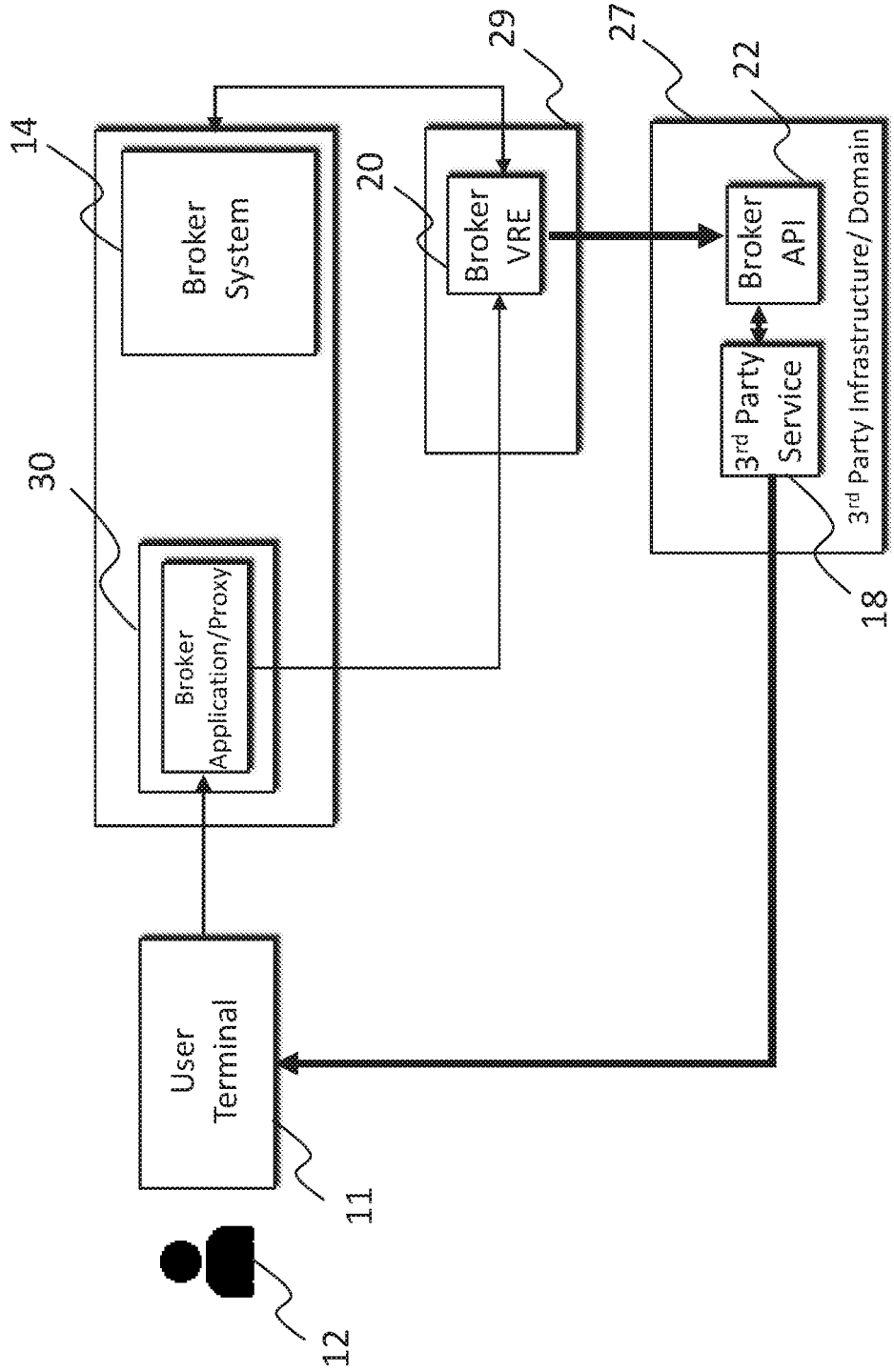
14
Broker System
20
Broker VRE
29
27
22
Broker API
3rd Party Service
3rd Party Infrastructure/ Domain
18
30
Broker Application/Proxy
User Terminal
11
12

Fig. 11

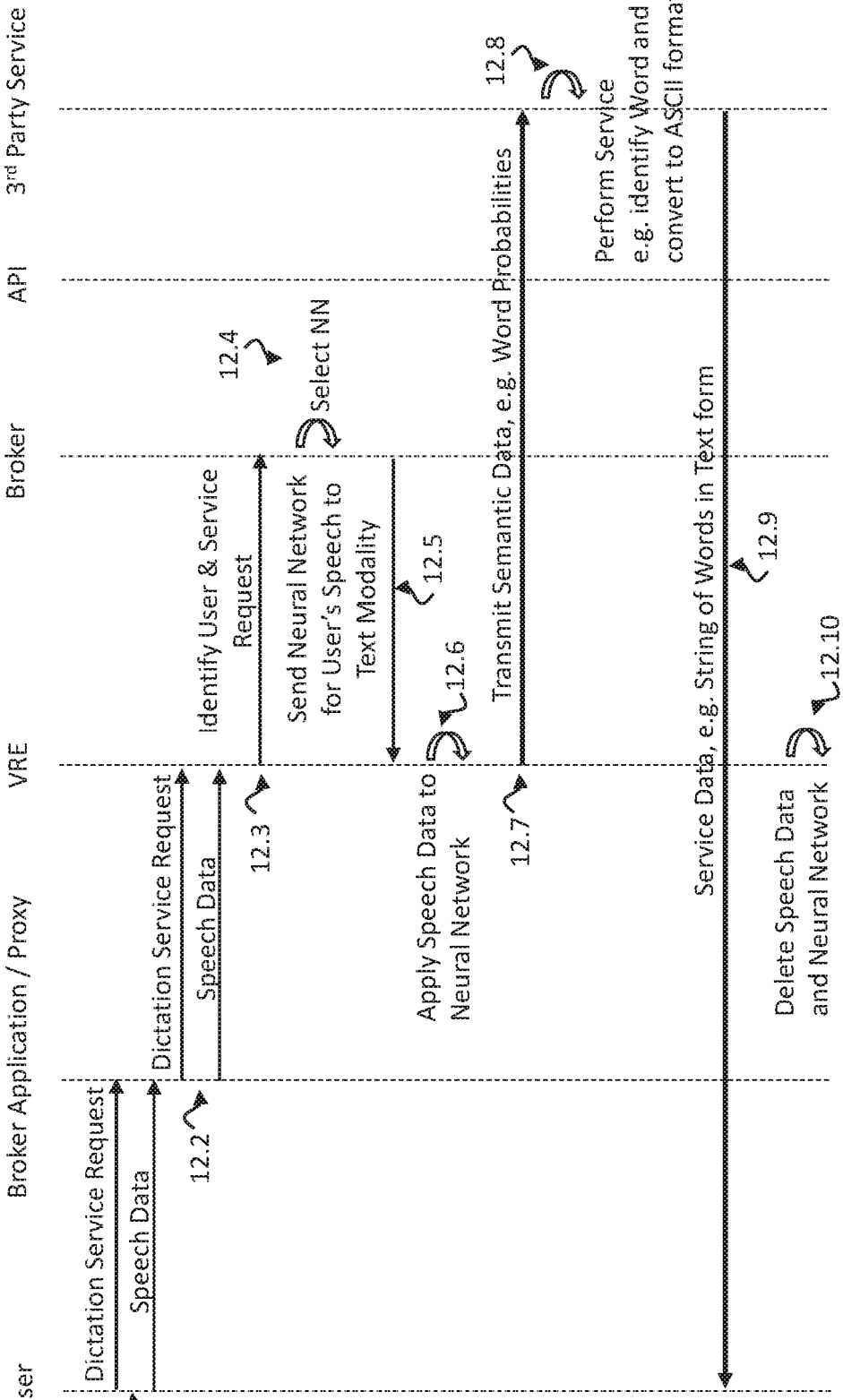
User
Broker Application / Proxy
VRE
Broker
API
3rd Party Service
Dictation Service Request
Speech Data
12.1
12.2
Dictation Service Request
Speech Data
12.3
Identify User & Service Request
12.4
Select NN
Send Neural Network for User's Speech to Text Modality
12.5
12.6
Apply Speech Data to Neural Network
12.7
Transmit Semantic Data, e.g. Word Probabilities
12.8
Perform Service e.g. identify Word and convert to ASCII format
Service Data, e.g. String of Words in Text form
12.9
Delete Speech Data and Neural Network
12.10

Fig. 12

Learned Model Data Processing

Field

This specification relates to a method and system for learned model data processing, for example where the learned model is an artificial neural network.

Background

Learned models are used for various computational tasks. For example, it is known to receive user data, such as speech, image or biometric data such as fingerprints, and to apply this data to a trained neural network to obtain a result. For example, the neural network may be trained to recognise words within an audio file or stream. Thereafter, in an operational phase, the neural network responds to receiving audio containing speech to produce information from which can be determined the words spoken, e.g. for conversion into text or commands.

In this context, speech analysis and voice services are gaining in popularity and are being provided by well-known service providers, for example for receiving and recognising speech received over a network from a smartphone, e.g. for issuing commands on the smartphone or search requests via the Internet. Another example is a dictation service, where speech is converted at a network service into text and sent back to the user device.

Summary

According to one aspect, there is provided a method comprising: at a first processing system, storing data representing a learned model for each of one or more users, the learned model being configured to produce output data responsive to user input data; and responsive to receiving a user request for a network-based service, sending the learned model for the user from the first processing system to an associated virtual processing environment provided in an external second processing system for temporary storage of the neural network for use in generating output data for subsequently providing to a third party service provider.

The method may further comprise sending, or causing to be sent, an application program to a user terminal, which application program, when executed at the user terminal, is configured to send the or each learned model from the user terminal to the first processing system over a secure channel.

The method may further comprise sending, or causing to be sent, an application program to the user terminal, which application program, when executed at the user terminal, is configured to send training data to the first processing system over a secure channel, and thereafter creating at the first processing system the or each learned model using the training data.

The method may further comprise deleting the training data from the first processing system subsequent to producing the learned model.

The method may further comprise sending, or causing to be sent, an application program to the user terminal, which application program, when executed at the user terminal, is configured to send user service requests and user input data to a predetermined location over a secure channel.

The application program may be configured to send the user service requests and user input data to a predetermined proxy system associated with the first processing system for re-direction to the virtual processing environment for requesting the learned model from the first processing system and applying the user input data to the learned model.

The application program may be further configured to send authentication data, the first processing system sending the learned model to the virtual processing environment only if the authentication data is received from the virtual processing environment.

The virtual processing environment may be configured in use to prevent external access to the user input data and the learned model.

The virtual processing environment may be configured to delete the user input data and the learned model subsequent to generating the output data.

The external second processing system may be an intermediary system between the first processing system and a service provider system.

The external second processing system may be within the third party service provider's domain or network infrastructure.

The learned model may be created using biometric training data, e.g. speech, a fingerprint or an image.

A plurality of learned models may be stored for the or each user, each learned model for a user being created using a different set of biometric training data.

A plurality of learned models may be stored for the or each user, each learned model for a user being associated with a particular third party service.

The method may further comprise providing a virtual processing environment for accessing the or each learned model stored on the first processing system, which virtual processing environment performs: receiving user input data from a user terminal; retrieving a learned model from the first processing system; processing the user input data with the learned model to generate output data; and providing the output data to a third party service through an output interface.

A further aspect of the invention provides a method comprising: at a processing system, providing a virtual processing environment for accessing one or more learned models stored on an external further processing system, which virtual processing environment performs: receiving user input data from a user terminal; retrieving a learned model from the external first processing system; processing the user input data with the learned model to generate output data; and providing the output data to a third party service through an output interface.

The method may further comprise receiving a user service request from the user terminal, and retrieving the learned model based on the user service request.

The method may further comprise deleting the learned model subsequent to providing the output data to the third party service.

The user input data and/or user service request may be received over a secure channel and the learned model is received over a secure channel.

The virtual processing environment may be provided in a third party domain or infrastructure.

The virtual processing environment may be configured in use to prevent external access to the user input data and the learned model by the third party domain or infrastructure.

According to a further aspect, there is provided means for performing the method of any preceding definition.

According to a further aspect, there is provided a computer program comprising instructions that when executed by a computer apparatus control it to perform the method of any preceding definition.

According to a further aspect, there is provided an apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured with the processor to cause the apparatus to: at a first processing system, store data representing a learned model for each of one or more users, the learned model being configured to produce output data responsive to user input data; and responsive to receiving a user request for a network-based service, send the learned model for the user from the first processing system to an associated virtual processing environment provided in an external second processing system for - temporary storage of the neural network for use in generating output data for subsequently providing to a third party service provider.

According to a further aspect, there is provided a non-transitory computer-readable storage medium having stored thereon computer-readable code, which, when executed by at least one processor, causes the at least one processor to perform a method, comprising: at a first processing system, storing data representing a learned model for each of one or more users, the learned model being configured to produce output data responsive to user input data; and responsive to receiving a user request for a network-based service, sending the learned model for the user from the first processing system to an associated virtual processing environment provided in an external second processing system for - temporary storage of the neural network for use in generating output data for subsequently providing to a third party service provider.

According to a further aspect, there is provided an apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured with the processor to cause the

apparatus to: at a processing system, provide a virtual processing environment for accessing one or more learned models stored on an external further processing system, which virtual processing environment performs: receive user input data from a user terminal; retrieve a learned model from the external first processing system; process the user input data with the learned model to generate output data; and provide the output data to a third party service through an output interface.

According to a further aspect, there is provided a non-transitory computer-readable storage medium having stored thereon computer-readable code, which, when executed by at least one processor, causes the at least one processor to perform a method, comprising: at a processing system, providing a virtual processing environment for accessing one or more learned models stored on an external further processing system, which virtual processing environment performs: receiving user input data from a user terminal; retrieving a learned model from the external first processing system; processing the user input data with the learned model to generate output data; and providing the output data to a third party service through an output interface.

Brief Description of the Drawings

Embodiments will now be described, by way of non-limiting example, with reference to the accompanying drawings, in which:

Figure 1 is a schematic diagram of a neural network topology;
Figure 2 is a flow diagram illustrating stages of training and using a neural network;
Figure 3 is an example of a block diagram of a data network in accordance with an embodiment;
Figure 4 is an example of a block diagram of a data network in accordance with a further embodiment;
Figure 5 is an example of a block diagram of part of the Figure 3 or Figure 4 data network, showing how a proxy server may be used in some embodiments;
Figure 6 is a flow diagram illustrating processing stages in a training stage of a neural network;
Figures 7a and 7b are examples of schematic views showing components of, respectively, a broker and virtual machine system used in the Figure 3 and Figure 4 data networks;
Figure 8 is a tabular view of an example of a neural network database within the broker system of Figure 7;

Figure 9 is an example of a block diagram of the Figure 3 data network for indicating data flow when the network is used to retrieve neural networks for providing a service to a user, in accordance with embodiments;

Figure 10 is an example of a sequence diagram showing data flow between systems of the Figure 9 data network;

Figure 11 is an example of a block diagram of the Figure 4 data network for indicating data flow when the network is used to retrieve neural networks for providing a service to a user, in accordance with embodiments; and

Figure 12 is an example of a sequence diagram showing data flow between systems of the Figure 9 or Figure 11 data network in relation to a speech to text dictation service.

Detailed Description of Embodiments

Embodiments herein relate to learned models, for example artificial neural networks. More particularly, embodiments relate to the creation, storage and distribution of learned models, which can be any learned model, but in particular the following focusses on neural networks.

In some embodiments, the learned models are personalised to particular users based on a machine learning phase. In some embodiments, the learned models may additionally, or alternatively, be particular to a given modality or service. For example, the machine learning phase may take as input a user's speech, image or fingerprint. These are examples of so-called biometric data.

In some embodiments, the learned models may be used for performing speech recognition, i.e., to predict the words spoken by a user in a received audio stream or file, with the resulting semantic data being used by a service provider to provide a service, for example to convert speech into text in a voice dictation system. Other examples include using the semantic data to issue commands to a computer or mobile terminal, or to a cloud-based system, or to select songs or telephone numbers from a list. Embodiments need not be restricted to biometric learned models, however.

Embodiments described herein can avoid problems with known systems which may collect a user's data in the training or learning phase, sometimes without consent. In such systems, a user's speech data may for example be recorded, compressed and then sent to the cloud or a server for analysis and neural network processing. Service providers may keep this data and use it for other purposes without the user's

knowledge. This may even be considered a violation of privacy and unauthorised use of personal data. Users not wishing to provide such personalised data may not therefore be able to use services which employ learned models, which may restrict their access to even fairly basic services now and in the future.

An artificial neural network (hereafter "neural network") is an example of a computational model with an input and an output and which performs a certain computational task. The input to the neural network may be represented by the input data on which to perform the task, or by features extracted from the input data. The output may be the desired task, or an intermediate result which may enable a subsequent module or modules to perform the task.

There are two main categories of neural networks, namely discriminative and generative. Discriminative networks allow for obtaining information about the posterior probability of the underlying factors of variations (such as classes) given the input data. Examples of such networks are convolutional neural networks, used for performing classification. The input may be raw data, or features extracted from raw data, and the output may be the estimated class. Generative networks allow for obtaining information about the joint probability distribution of the data and the underlying factors of variation, or about the likelihood of the data given factors of variation. These neural networks are usually used either for pre-training a classification model, or for generating artificial data such as artificial images, or for de-noising and reconstructing data.

Referring to Figure 1, which is a generalised neural network 1, the network is usually represented by a series of units, sometimes referred to as neurons, which may perform a basic computation. In the example shown, these units are input units 3, intermediate units 4, and output units 5 which are connected. The units 3, 4, 5 are connected in specific ways. Usually, the units 3, 4, 5 are organised in layers as shown, and units within each layer are not connected. Instead, units of different layers are connected to each other. The connections can be dense, and the layer may in this case be called fully-connected, or sparse, such as for convolutional layers.

A neural network may be defined by two sets of numerical values: hyper-parameters (or topology) and parameters (or weights).

The hyper-parameters are values which may be set by the human practitioner based on his or her experience and thus they are usually not learnt. They define the general structure of the network, such as number of layers, the number of units per layer, the type of connection (dense or sparse), activation functions, etc. There is some research ongoing for learning the hyper-parameters from training data, but the most common method is to test out values within pre-defined ranges, i.e. by using grid-search methods. The present embodiments do not depend on the specific way by which hyper-parameters are set.

The parameters are the weights of a neural network and their values are learnt from data. They may represent the connection strength between units. The parameters are set during a so-called training phase in an iterative manner by running an optimization routine on training data. This optimization usually consists of finding the combination of weights for which the objective function is optimized. The objective function can be a classification-error function, and thus the optimization may be a minimization problem. For example, one basic minimization routine is a Stochastic Gradient Descent (SGD) routine, which uses the gradient of the objective function (computed for example by the Back-propagation algorithm, Monte-Carlo methods or Reinforcement learning techniques) for updating the weights.

Once the hyper-parameters and weights have been set, the neural network is completely defined by those two types of data, and it is ready for deployment. References made hereinafter to "neural network" may refer to these types of data, or subsets thereof.

Figure 2 shows the two main phases of neural network construction and use, namely a training phase 7 and an operating or inference phase 8. The feedback arrow indicates that re-training may optionally be performed to further refine or optimise the neural network 1.

As a neural network 1 is trained on training data, its final form is dependent on the training data. Thus, the quality of a model may be modified by modifying the training data and re-running the training phase 7. The training data may implicitly express a so-called "empirical distribution" which is a noisy estimate of the true data distribution (as the training data is a sub-set of the whole existing data). This empirical distribution may not represent all the possible data. Thus, it may be better to fine-tune the model

on a specific domain or test use-case. In the case of a learned model for speech recognition (e.g. for use in a speech-to-text service) this may mean that a generic model trained on a number of users may be improved for another user (who did not provide some of the initial training data) when fine-tuned on that specific user.

It will be appreciated that speech analysis and voice services are becoming popular and being used by large scale service providers such as Google Inc.'s Google Now service, Apple Inc.'s Siri service, Microsoft Corporation's Cortana service and Amazon.com Inc.'s Alexa service. All are Registered Trade Marks (RTMs). These services constantly improve and train their neural networks in order to be more robust. Most of these large-scale systems work in a speaker-independent fashion, where the user's voice data may be used to train a global neural network.

Some services are used for dictation. In such systems (such as in Nuance Communications, Inc.'s Dragon (RTM) speech recognition software) speaker dependent training may be performed in order to reduce dictation error and increase system robustness.

Thus, the following embodiments focus on methods and systems for speech recognition using neural networks, but it should be understood that the methods and systems are applicable to any construction and distribution of learned models.

In overview, the embodiments provide for the provisioning of neural networks (in terms of one or both of the learning and inference phases 7, 8) which improves the protection given to personal data provided by the user, e.g. a person or device providing personal user input data to use a service, such as a voice service. In some embodiments, this user input data may be biometric data such as voice data, fingerprint data or image data, and may be considered sensitive and private.

The embodiments provide a broker system which has a trust relationship with the user. During the learning or training phase 7, the broker system may be used to train a neural network for the user, e.g. for speech recognition. The training data may then be deleted by the broker system when the neural network is created. By deleting, this may include any means of making the original data unreadable and unrecoverable. During the inference phase 8, the broker system may provide the resulting neural network on an on-demand basis to an associated virtual machine (which may be, for example, a virtual

runtime environment (VRE) which generates for a third party service provider semantic output data for use in providing their service. The neural network itself may not be made directly available to the third party service provider, thereby offering protection of sensitive data. Similarly, received user input data which is to be processed with the neural network may be received by the virtual machine (e.g. a virtual runtime environment) and may not be made available to the third party service provider. Both the neural network and the user input data may be deleted from the virtual machine after the processing result is provided to the third party service provider, meaning that it is temporarily held external to the broker system, improving privacy and security of personal data. Again, deleting may comprise any means of making the original data unreadable.

The semantic output data may be a set of words identified by applying, or processing, received user's speech with their trained neural network. The words may be used by the third party service provider to deliver a service, e.g. returning the words to the user's terminal or mobile handset as a dictation result, or for controlling some aspect of their terminal such as issuing commands, selecting music tracks and/or selecting a telephone number from a list of contacts, to give some examples.

In the following, a user is a person, entity or system that wishes to use or consume a third party service offered by an external third party service provider. A user terminal is any user-end processing system, such as a computer, laptop, smartphone or tablet, to give some examples.

In the following, a broker or broker system is hardware, software or a combination of both that is external to the user and the third party service provider. The broker system may be used in the learning and the inference phases 7, 8, although in some embodiments the learning phase may be performed at a user terminal. The broker system comprises a server which trains, stores and sends on an on-demand basis neural networks to an external virtual machine. The broker system may have a trust relationship with the user, which may be established by a registration process, and may use one or both of software application(s) and/or a proxy system for managing access and consent management with users.

For example, the broker system may have the permission to deploy software to one or more user terminals in order to access user input data during the training and inference

phases 7, 8 and to receive service requests. For example, the broker system may send a software application to one or more user terminals which in the training phase 7 requests training data which is used at the broker system to generate a personalised neural network for storage. For example, the broker system may send a software application which in the training phase 7 performs training at the user terminal itself so that the personalised neural network is generated at the user's terminal and then sent back to the broker system for storage. In some examples, the training may be performed in a cloud-based system in which case the broker system may send a webpage which provides user access to the said cloud-based system which generates the neural network and sends it to the broker system when complete. The same, or a different, software application provided by the broker may be used in the inference phase 8 to send service requests for using one or more third party services. The broker system may provide the software application(s) directly or cause them to be sent from a remote or external location.

In some embodiments, the broker system and/or the above described software establishes a secure end-to-end channel between itself and the user terminal, for example using a HTTPS connection, a secure TCP connection with SSL, a secure VPN, a secure API call, to give some examples. Where a broker proxy system is used between the user terminal and the broker servers, a secure channel may be set up between the user terminal and the proxy system, and between the proxy system and the user terminal.

In some embodiments, the broker provides a virtual machine to an external system, which may be a third party service provider or an intermediary service.

In the following, a virtual machine is a virtual processing system (or virtual processing environment) external to the broker system but which is associated with the broker system, e.g. by having a trust relationship with the broker system and/or being managed or controlled by the broker system. For example, the virtual machine may be a virtual runtime environment (VRE) which may be any virtual runtime environment, such as the Java Virtual Machine. The virtual machine may be deployed to the external system by the broker system in some embodiments. The virtual machine may be a software emulation of a computer system provided and controlled by the broker, using processors (including processor circuitry) and memory provided at the external system, but which are not directly accessible by other parts of the external system. The virtual

machine may use an application program interface (API) to control which data is made available to other processing systems, for example a third party service provider. A service request is a data message identifying a service required by the user, for example a speech to text service, a speech to command service, a fingerprint recognition service etc. The virtual machine may receive service requests from the user terminal, and may use this information to request an appropriate neural network from the broker system. The virtual machine may need to authenticate the server request with the broker system for this purpose. When the neural network is received by the virtual machine, user input data may be applied to the neural network to produce semantic output which is delivered to, e.g. the third party service provider, through the API.

The virtual machine may delete any personalised data, e.g. the retrieved neural network and/or the user data, after the semantic output is produced and/or sent to the third party service provider. Therefore, the personalised data is stored only temporarily.

In some embodiments, the broker may use the user data to perform one or more optimisation tasks, which may be with the prior approval of the user, to improve the neural network(s). In this case, the personalised data may be sent to the broker system and is deleted by the broker subsequently.

The virtual machine may be configured to establish a secure end-to-end channel between the user terminal and/or a broker proxy, if used, and the broker system. For example, the secure channel may use a HTTPS connection, a secure TCP connection with SSL, a secure VPN, a secure API call, to give some examples.

In some embodiments, the virtual machine may be located in the domain or infrastructure of a third party service provider. In some embodiments, the virtual machine may be located in the domain or infrastructure of an intermediary inference service (IIS).

Possible deployment options for implementing a virtual machine or virtual processing environment at an external system are now briefly described.

For example, a specialised architecture for isolating data and processes running on the same or distributed hardware may be used to provide the virtual machine; the third party service provider may reside in in the same architecture, although isolated

therefrom in separate private spaces. One such example is Heroku.com's private space service.

For example, separate, dedicated broker-owned hardware may be provided inside the third party service provider's domain, and a virtualized environment may be used to deploy the virtual machine in that hardware. As the hardware is owned and managed by the broker system, the only way to connect to it is via a secure IP communication channel (e.g. the above-mentioned API) for the third party service provider. A secure VPN could also be established between the broker system and the virtual machine for application life-cycle management and on-demand neural network retrieval.

For example, the virtual machine may be deployed as an in-memory process. This may require a dedicated and separate in-memory running process to launch the virtual machine when needed. This option may provide fast performance although may require another secure memory access and process controller between the third party service provider's system and the virtual machine to ensure that other systems cannot see and access the contents of the process and the memory.

For example, a separation kernel may be used to deploy the virtual machine to the third party service provider's domain, e.g. see https://en.wikipedia.org/wiki/Separation_kernel.

For example, a virtualization approach which is hardened to cover vulnerabilities for virtual machine memory and process access may be used so that any memory or process access attempts may be detected and avoided.

For example, encryption may be used in addition to the above-mentioned options for data storage and virtual processes running in the memory, as a security measure. For example, a VPN between the user terminal and the virtual machine can be used as an additional security measure. For example, the use of anti-tampering methods may be used to prevent and/or detect physical access to hardware running the virtual machine may be employed.

In the following, a third party service provider is a system or service external and/or remote to the user terminal which provides a service responsive to a user request, although it may not receive the user request directly. The third party service provider

may not have the user's consent to store or possess user data or derivatives thereof, e.g. their personalised neural network(s). The third party service provider may use semantic data or information derived from the virtual machine processing a neural network in response to a user request. For example, the third party service provider may receive semantic data or information from the above-mentioned virtual machine through an API, which API prevents access by the third party service provider to the user data and the neural network itself. The third party service provider may be network-based, meaning that they are external and/or remote to the user terminal but the user terminal can communicate over some data network, for example using the Internet or a local network. The data path between the user terminal and the third party service provider may be wired or wireless.

The third party service provider may receive the semantic data or information directly from an IIS which has the computational capabilities for running neural network inference on a large scale. Amazon.com Inc.'s AWS service is an example of an IIS, and the third party service may be an online shop, to give an example. Alternatively, the third party service provider may perform the inference at their domain(s) and/or infrastructure and provide the service to the user. Google Inc.'s Assistant service is one such example.

As the lifecycle of the virtual machine may be 'managed' by the broker system, the only communications between the virtual machine and the third party service provider may be the IP based communications for sending the semantic information or data over the defined API.

Embodiments will now be described in greater detail.

Referring to Figure 3, a first embodiment system 10 is shown in overview. It comprises, in a data network, a user terminal 11 which is associated with a user 12, a broker system 14 and a third party system 16. The user terminal 11 can be any user-end processing system, such as a computer, laptop, smartphone or tablet, to give some examples. The broker system 14 may comprise one or more computer systems external to the user terminal 11, providing processing and memory storage capability. The third party system 16 may comprise one or more computer systems external to the user terminal 11 and the broker system 14, for providing a service to users. The service may be, for example, a speech to text service whereby the user sends from the user terminal 11 a

data file or stream containing spoken audio over the network for receiving from the third party system 16 recognised text.

In the training phase 7 (see Figure 2) for generating neural networks, only the user terminal 11 and broker system 14 are needed, as will be described later on. The third party system 16 is used during the inference phase 8.

The third party system 16 comprises a third party service module 18 which may be software and/or hardware configured to perform a service responsive to received semantic input from a VRE. The semantic input is received via a defined broker API 22.

As mentioned above, the VRE 20 is associated with, and managed by, the broker system 14. The VRE 20 is configured to receive user service requests (e.g. a command requesting use of the third party speech to text service) and user data (e.g. speech.) The VRE 20 then requests an appropriate neural network from the broker system 14. This may involve first authenticating the user service request to the broker system 14 using identification data, which the broker system uses to select the appropriate neural network. This selection may also depend on the type of service being used. Upon receiving the neural network, the VRE 20 then processes the user data with the neural network to generate the semantic output, and may then delete the user data and the user service request.

The broker API 22 provides the functions and procedures that allows only certain information to be sent from the VRE 20 to the third party service module 18. For example, only the semantic data from the VRE 20 may be received. The third party service module 18 may then provide the service, e.g. with additional processing or formatting, to the user terminal 11. The VRE 20 may be implemented using any of the above methods for partitioning or separating it from third party system access other than through the API 22. In all embodiments, the API may be a web API or a classical software API.

Referring to Figure 4, a second embodiment system 25 is shown in overview. The difference in this system 25 is that a third party service provider system 27 does not have the VRE 20 within its domain or infrastructure. Rather, the VRE 20 is provided on an external IIS 29, for example an IIS which has the computational capabilities for

running neural network inference on a large scale. The VRE 20 communicates with the broker system 14 and the third party service module 18 in the same way, using the broker system's API 22 for the latter. However, no personalised data is accessible at the third party service provider system 27 and it only receives the semantic data for service provisioning to the user terminal 11.

Referring to Figure 5, in some embodiments, the broker system 14 may use a proxy 30 to receive and process the user service requests and user data.

Referring to Figure 6, a generalised method for performing the training phase 7 of a neural network is shown. The training phase 7 in overview comprises obtaining training data from the user, for example via the user terminal 11, and using known methods of generating a neural network. In a first step 6.1 the training phase is commenced, which may include sending an application or link to a webpage to prompt the user to provide the training data and/or to indicate or select the type of training data (i.e. the modality) and/or a particular service with which the training data is to be associated, e.g. a speech-to-text service, a fingerprint recognition service etc. In step 6.2 the raw training data is received from the user, e.g. speech. In step 6.3 the neural network is trained and in step 6.4 is stored on a database of the broker system 14. Once stored, in step 6.5 the raw data is deleted.

It should be noted that the training and generating of the neural networks can be performed either at the broker system 14 or at the user terminal 11. In this respect, the broker system 14 may send an application program to the user terminal 11 which, when executed at a processor (including processor circuitry) on the user terminal, receives and generates the neural network from the raw data. It is then only the neural network, and not the raw data, that is sent to the broker system 14. Alternatively, the broker system 14 may send an application program to the user terminal 11 which transmits the raw data to the broker system which itself generates the neural network and then deletes the raw data. In other embodiments, the broker system 14 may send a link to a cloud-based service which receives the raw data and generates the neural network.

In some embodiments, when training is performed at the broker system 14, instead of sending the raw user data, it can be sent in encoded form, e.g. using MPEG-4, AAC-LC encoded audio formats.

In some embodiments, a secure channel is set-up between the user terminal 11 and the broker system 14 for the training phase, e.g. using a VPN or SSL, which can be encrypted.

If the user has an associated neural network (i.e. for the same modality/service) at the broker system 14, it may be updated and the previous one deleted.

Figure 7a shows an example schematic diagram of components of the broker system 14. The broker system 14 has a controller 32, a memory 34, RAM 36 and a neural network database 38. The controller 32 is connected to each of the other components in order to control operation thereof.

The memory 34 may be a non-volatile memory such as read only memory (ROM) a hard disk drive (HDD) or a solid state drive (SSD). The memory 34 stores, amongst other things, an operating system 39 and software applications 40. The RAM 36 is used by the controller 32 for the temporary storage of data. The operating system 39 may contain code which, when executed by the controller 32 in conjunction with RAM 36, controls operation of each of hardware components of the terminal.

The controller 32 may take any suitable form. For instance, it may be a microcontroller, plural microcontrollers, a processor (including processor circuitry), or plural processors (each including processor circuitry).

The software applications 40 may provide the functionality for the training and inference phase operations mentioned herein. For example, the software applications 40 may include training application which are to be deployed to the user terminal 11 via the proxy 30 in the training phase, as mentioned above, for receiving and storing personalised neural networks for users on the neural network database 38. For example, the software applications may handle the control and interaction with the VRE 20 for authenticating user service requests and delivering requested neural networks to the VRE in the inference phase.

The neural network database 38 can be any form of non-volatile memory for storing neural networks for different users.

Figure 8 shows a schematic representation of the neural network database 38. It will be seen that for individual users, one or more personalised neural networks may be stored in relation to different modalities, e.g. speech, fingerprints, images, and in relation to different tasks or services, e.g. a voice to text service, a voice to song service etc. It will be appreciated therefore that the stored neural networks may be personalised and specific to individual users rather than being globally generated.

For completeness, Figure 7b shows an example schematic diagram of components of the hardware and/or software providing the VRE 20. It will be appreciated that the shown hardware may be within a third party infrastructure or domain, e.g. an IIS 16 or third party system 27, but the shown components are partitioned and private from that domain. The VRE 20 has a controller 42, a memory 44, and RAM 46. The controller 46 is connected to each of the other components in order to control operation thereof.

The memory 44 may be a non-volatile memory such as read only memory (ROM) a hard disk drive (HDD) or a solid state drive (SSD). The memory 40 stores, amongst other things, an operating system 49 and software applications 50. The memory 40 also stores the abovementioned API 22 which defines which information is made available to third parties. The RAM 46 is used by the controller 42 for the temporary storage of data. The operating system 49 may contain code which, when executed by the controller 42 in conjunction with RAM 46, controls operation of each of hardware components of the terminal.

The controller 42 may take any suitable form. For instance, it may be a microcontroller, plural microcontrollers, a processor (including processor circuitry), or plural processors (each including processor circuitry).

The software applications 50 may provide the functionality for the receiving of user data, e.g. private data, and user service requests, from user terminals, and the requesting and receiving of neural networks from the broker system 14. The software applications 50 may handle authentication of user service requests and the establishment of secure data channels.

Referring now to Figures 9 and 10, the operation of the various systems during the inference phase will be described. Figure 9 is similar to Figure 3 but has been adapted to indicate the data channels used during the inference phase. It is assumed in this

case that the broker system 14 operates in conjunction with an associated proxy server 30 which runs software processes to be explained below. The proxy server 30 is in a secure and trusted relationship with the broker system 14. However, the use of a proxy server 30 is not essential. Figure 10 is a sequence diagram showing the method steps performed at each system module. It is assumed that the user 12 has already provided via the training phase one or more neural networks which are stored at the broker system 14.

Initially, in step 10.1 the user 12 wishing to use the third party service provider's service 18 sends (i) a user service request and (ii) user data to the proxy server 30. This happens over a secure IP based channel, e.g. using HTTPS, secure TCP with SSL, VPN a secure API call etc.

The user service request may include the identity of the user and, in some embodiments, authentication information, as well as the service needed, e.g. by specifying that a voice-to-text service is requested and/or the identity of the third party service provider 16 which is to provide the service. The user data may be raw biometric data, for example an audio file or stream containing speech.

Next, in step 10.2, the proxy server 30 redirects the user service request and the user data to the VRE 20, which may be over a secure channel as mentioned above. This may be performed in a secure manner, in order that it is transparent to the third party service provider 18 in the sense that they cannot access the user data. This may involve configuration of the VRE 20 and proxy server 30, e.g. by IP address or VPN access, so that the user data flows directly to the VRE.

Next, in step 10.3, the VRE 20 sends a request to the broker system 14 for the user's neural network. This may involve first transmitting the user's identity and authentication information, which the broker system 14 may use to verify that the user 12 has authorised the VRE to access their neural network. If verified, in step 10.4 the information in the service request is used by the broker system 14 to select the required neural network for the user. The selected neural network is then sent by the broker system 14 to the VRE 20 in step 10.5. Such processes again occur over a secure channel.

In step 10.6, at the VRE 20, the received neural network is stored and processed with the user data to generate a result, which will be the semantic data.

In step 10.7, the VRE 20 sends the semantic data via the API 22 to the third party service module 18 where it may be processed and/or formatted in accordance with said module's operation (step 10.8). Having performed the service on the semantic data, the third party service module 18 sends the service data to the user terminal 11 (step 10.9).

The VRE 20 may then delete the user data and the user service request from its own memory in step 10.10. This may be performed after the semantic data is generated. In some embodiments, the user data may be retained however, if the user consents to it being used to retrain one or more of their neural networks. In such cases, the user data is deleted subsequently, and so is only temporarily held at the VRE 20. According to an embodiment, the user data may be retained for a predetermined time period, or, until the user is determined not to use the third-party service anymore.

For illustration purposes in Figures 9 and 10, all data communications performed using the thin arrows indicate that a secure data channel is set up under the control of the broker system 14 or software associated with, or derived therefrom. The wider arrows indicate the non-private, semantic data.

Referring now to Figure 11, which is similar to Figure 4, the data channels used during the inference phase are indicated in the case where the VRE 20 is provided within the IIS 29 external to the third party service provider 27. The process employed in this case is similar to that shown in Figure 10.

Figure 12 is a sequence diagram showing the method steps performed at each system module in either of the Figure 9 or 11 systems, specifically for a network-based dictation (speech to text) service. It is assumed that the user 12 has already provided via the training phase one or more neural networks which are stored at the broker system 14 and which are specific to the user and the dictation service.

In a first step 12.1 the user 12 wishing to use the dictation service 18 sends (i) a dictation service request and (ii) speech data to the proxy server 30. This happens over a secure channel, e.g. using HTTPS, secure TCP with SSL, VPN a secure API call etc.

The dictation service request may include the identity of the user and, in some embodiments, authentication information, as well as the service needed, e.g. by specifying that a dictation service is requested and/or the identity of the dictation service provider 16.

Next, in step 12.2, the proxy server 30 redirects the dictation service request and the speech data to the VRE 20, which may be over a secure channel as mentioned above. This may be performed in a secure manner, in order that it is transparent to the third party service provider 18 in the sense that they cannot access the speech data. This may involve configuration of the VRE 20 and proxy server 30, e.g. by IP address or VPN access, so that the speech data flows directly to the VRE.

Next, in step 12.3, the VRE 20 sends a request to the broker system 14 for the user's neural network. This may involve first transmitting the user's identity and authentication information, which the broker system 14 may use to verify that the user 12 has authorised the VRE to access their neural network. If verified, in step 12.4 the information in the dictation service request is used by the broker system 14 to select the required speech-to-text neural network which has been personalised for the user in the learning phase. The selected neural network is then sent by the broker system 14 to the VRE 20 in step 12.5. Such processes again occur over a secure channel.

In step 12.6, at the VRE 20, the received neural network is stored and processed with the user data to generate a result, which will be the semantic data e.g. the words and/or word probabilities recognised from the speech.

In step 12.7, the VRE 20 sends the semantic data via the API 22 to the dictation service module 18 where it may be processed and/or formatted in accordance with said module's operation (step 12.8) to produce service data. Having performed the service on the semantic data, the dictation service module 18 sends the service data to the user terminal 11 (step 12.9).

The VRE 20 may then delete the speech data and the dictation service request from its own memory in step 12.10. This may be performed after the semantic data is generated. In some embodiments, the user data may be retained however, if the user consents to it being used to retrain one or more of their neural networks. In such cases, the user data is deleted subsequently, and so is only temporarily held at the VRE 20.

In summary, there is disclosed methods and systems for learning and inference stages using one or more personalised neural networks. Neural networks have been described as one example of learned models, and other such models may be learned and deployed along the same principles. Embodiments provide a brokerage method, whereby a trusted relationship between a broker and the end-user may be established and the broker may store the personalised neural networks for provisioning to an associated virtual machine stored in an external domain or infrastructure, on a temporary, on-demand basis. The virtual machine, which can be a VRE, may receive user service requests and private data, request and obtain the required neural network(s) from the broker, and process the private data with said retrieved neural network(s) to produce a semantic result for the service provider.

Advantages include, but are not limited to, the feature whereby no personal data need be stored at the broker system after learning has completed to produce the neural network. Neural networks may be unique to individual users, and may be updated from time-to-time, so that the most up-to-date neural network for a user and/or modality and/or service type is stored. Only semantic data may be provided to third party service providers through a predefined API which is set up and controlled by the broker system. The neural networks are only provided external to the broker system temporarily, and on-demand. Overall, advantages may be provided in terms of improving user confidence in using such external, cloud-based third party services for acting on personal data such as biometric data.

It will be appreciated that the above described embodiments are purely illustrative and are not limiting on the scope of the invention. Other variations and modifications will be apparent to persons skilled in the art upon reading the present application.

Moreover, the disclosure of the present application should be understood to include any novel features or any novel combination of features either explicitly or implicitly disclosed herein or any generalization thereof and during the prosecution of the present application or of any application derived therefrom, new claims may be formulated to cover any such features and/or combination of such features.

Claims

1. A method comprising:

at a first processing system, storing data representing a learned model for each of one or more users, the learned model being configured to produce output data responsive to user input data; and

responsive to receiving a user request for a network-based service, sending the learned model for the user from the first processing system to an associated virtual processing environment provided in an external second processing system for temporary storage of the neural network for use in generating output data for subsequently providing to a third party service provider.

2. The method of claim 1, further comprising sending, or causing to be sent, an application program to the user terminal, which application program, when executed at the user terminal, is configured to send training data to the first processing system over a secure channel, and thereafter creating at the first processing system the or each learned model using the training data.

3. The method of claim 2, further comprising deleting the training data from the first processing system subsequent to producing the learned model.

4. The method of claim 1, further comprising sending, or causing to be sent, an application program to a user terminal, which application program, when executed at the user terminal, is configured to send the or each learned model from the user terminal to the first processing system over a secure channel.

5. The method of any preceding claim, further comprising sending, or causing to be sent, an application program to the user terminal, which application program, when executed at the user terminal, is configured to send user service requests and user input data to a predetermined location over a secure channel.

6. The method of claim 5, wherein said application program is configured to send the user service requests and user input data to a predetermined proxy system associated with the first processing system for re-direction to the virtual processing environment for requesting the learned model from the first processing system and applying the user input data to the learned model.

7. The method of claim 5 or claim 6, wherein the application program is further configured to send authentication data, the first processing system sending the learned model to the virtual processing environment only if the authentication data is received from the virtual processing environment.

8. The method of any preceding claim, wherein the virtual processing environment is configured in use to prevent external access to the user input data and the learned model.

9. The method of any preceding claim, wherein the virtual processing environment is configured to delete the user input data and the learned model subsequent to generating the output data.

10. The method of any preceding claim, wherein the external second processing system is an intermediary system between the first processing system and a service provider system.

11. The method of any of claims 1 to 9, wherein the external second processing system is within the third party service provider's domain or network infrastructure.

12. The method of any preceding claim, wherein the learned model is created using biometric training data, e.g. speech, a fingerprint or an image.

13. The method of claim 12, wherein a plurality of learned models are stored for the or each user, each learned model for a user being created using a different set of biometric training data.

14. The method of any preceding claim, wherein a plurality of learned models are stored for the or each user, each learned model for a user being associated with a particular third party service.

15. The method of any preceding claim, further comprising providing a virtual processing environment for accessing the or each learned model stored on the first processing system, which virtual processing environment performs:

receiving user input data from a user terminal;

retrieving a learned model from the first processing system;

processing the user input data with the learned model to generate output data; and

providing the output data to a third party service through an output interface.

16. A method comprising:

at a processing system, providing a virtual processing environment for accessing one or more learned models stored on an external further processing system, which virtual processing environment performs:

receiving user input data from a user terminal;

retrieving a learned model from the external first processing system;

processing the user input data with the learned model to generate output data; and

providing the output data to a third party service through an output interface.

17. The method of claim 16, further comprising receiving a user service request from the user terminal, and retrieving the learned model based on the user service request.

18. The method of claim 16 or claim 17, further comprising deleting the learned model subsequent to providing the output data to the third party service.

19. The method of any preceding claim, wherein the user input data and/or user service request is received over a secure channel and the learned model is received over a secure channel.

20. The method of any preceding claim, wherein the virtual processing environment is provided in a third party domain or infrastructure.

21. The method of claim 20, wherein the virtual processing environment is configured in use to prevent external access to the user input data and the learned model by the third party domain or infrastructure.

22. A computer program comprising instructions that when executed by a computer apparatus control it to perform the method of any preceding claim.

23. Apparatus configured to perform the method of any of claims 1 to 21.

24. Apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured with the processor to cause the apparatus to:

at a first processing system, store data representing a learned model for each of one or more users, the learned model being configured to produce output data responsive to user input data; and

responsive to receiving a user request for a network-based service, send the learned model for the user from the first processing system to an associated virtual processing environment provided in an external second processing system for temporary storage of the neural network for use in generating output data for subsequently providing to a third party service provider.

25. Apparatus of claim 24, wherein the at least one memory and the computer program code are configured with the processor to further cause the apparatus to send, or cause to be sent, an application program to the user terminal, which application program, when executed at the user terminal, is configured to send training data to the first processing system over a secure channel, and thereafter create at the first processing system the or each learned model using the training data.

26. Apparatus of claim 25, wherein the at least one memory and the computer program code are configured with the processor to further cause the apparatus to delete the training data from the first processing system subsequent to producing the learned model.

27. Apparatus of claim 24, wherein the at least one memory and the computer program code are configured with the processor to further cause the apparatus to send, or cause to be sent, an application program to a user terminal, which application program, when executed at the user terminal, is configured to send the or each learned model from the user terminal to the first processing system over a secure channel.

28. Apparatus of any of claims 24 to 27, wherein the at least one memory and the computer program code are configured with the processor to further cause the apparatus to send, or cause to be sent, an application program to the user terminal, which application program, when executed at the user terminal, is configured to send user service requests and user input data to a predetermined location over a secure channel.

29. Apparatus of claim 28, wherein said application program is configured to send the user service requests and user input data to a predetermined proxy system associated with the first processing system for re-direction to the virtual processing environment for requesting the learned model from the first processing system and applying the user input data to the learned model.

30. Apparatus of claim 28 or claim 29, wherein the at least one memory and the computer program code are configured with the processor to further cause the apparatus to send authentication data, the first processing system sending the learned model to the virtual processing environment only if the authentication data is received from the virtual processing environment.

31. Apparatus of any of claims 24 to 30, wherein the virtual processing environment is configured in use to prevent external access to the user input data and the learned model.

32. Apparatus of any of claims 24 to 31, wherein the virtual processing environment is configured to delete the user input data and the learned model subsequent to generating the output data.

33. Apparatus of any of claims 24 to 32, wherein the external second processing system is an intermediary system between the first processing system and a service provider system.

34. Apparatus of any of claims 24 to 32, wherein the external second processing system is within the third party service provider's domain or network infrastructure.

35. Apparatus of any of claims 24 to 34, wherein the learned model is created using biometric training data, e.g. speech, a fingerprint or an image.

36 Apparatus of claim 35, wherein a plurality of learned models are stored for the or each user, each learned model for a user being created using a different set of biometric training data.

37. Apparatus of any of claims 24 to 36, wherein a plurality of learned models are stored for the or each user, each learned model for a user being associated with a particular third party service.

38. Apparatus of any of claims 24 to 37, wherein the at least one memory and the computer program code are configured with the processor to further cause the apparatus to provide a virtual processing environment for accessing the or each learned model stored on the first processing system, which virtual processing environment performs:

receiving user input data from a user terminal;

retrieving a learned model from the first processing system;

processing the user input data with the learned model to generate output data; and

providing the output data to a third party service through an output interface.

39. A non-transitory computer-readable storage medium having stored thereon computer-readable code, which, when executed by at least one processor, causes the at least one processor to perform a method, comprising:

at a first processing system, storing data representing a learned model for each of one or more users, the learned model being configured to produce output data responsive to user input data; and

responsive to receiving a user request for a network-based service, sending the learned model for the user from the first processing system to an associated virtual processing environment provided in an external second processing system for - temporary storage of the neural network for use in generating output data for subsequently providing to a third party service provider.

40. The non-transitory computer-readable storage medium of claim 39, wherein the computer-readable code, when executed by at least one processor, further causes the at least one processor to send, or cause to be sent, an application program to the user

terminal, which application program, when executed at the user terminal, is configured to send training data to the first processing system over a secure channel, and thereafter create at the first processing system the or each learned model using the training data.

41. The non-transitory computer-readable storage medium of claim 40, wherein the computer-readable code, when executed by at least one processor, further causes the at least one processor to delete the training data from the first processing system subsequent to producing the learned model.

42. The non-transitory computer- readable storage medium of any of claims 39 to 41, wherein the computer-readable code, when executed by at least one processor, further causes the at least one processor to send, or cause to be sent, an application program to the user terminal, which application program, when executed at the user terminal, is configured to send user service requests and user input data to a predetermined location over a secure channel

43. Apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured with the processor to cause the apparatus to:

at a processing system, provide a virtual processing environment for accessing one or more learned models stored on an external further processing system, which virtual processing environment performs:

receive user input data from a user terminal;

retrieve a learned model from the external first processing system;

process the user input data with the learned model to generate output data; and

provide the output data to a third party service through an output interface.

44. Apparatus of claim 43, wherein the at least one memory and the computer program code are configured with the processor to cause the apparatus to receive a user service request from the user terminal, and to retrieve the learned model based on the user service request.

45. Apparatus of claim 44, wherein the at least one memory and the computer program code are configured with the processor to cause the apparatus to delete the learned model subsequent to providing the output data to the third party service.

46. Apparatus of any of claims 24 to 38 or 43 to 45, wherein the user input data and/or user service request is received over a secure channel and the learned model is received over a secure channel.

47. Apparatus of any of claims 24 to 38 or 43 to 46, wherein the virtual processing environment is provided in a third party domain or infrastructure.

48. Apparatus of claim 47, wherein the virtual processing environment is configured in use to prevent external access to the user input data and the learned model by the third party domain or infrastructure.

49. A non-transitory computer-readable storage medium having stored thereon computer-readable code, which, when executed by at least one processor, causes the at least one processor to perform a method, comprising:

at a processing system, providing a virtual processing environment for accessing one or more learned models stored on an external further processing system, which virtual processing environment performs:

receiving user input data from a user terminal;

retrieving a learned model from the external first processing system;

processing the user input data with the learned model to generate output data; and

providing the output data to a third party service through an output interface.

50. The non-transitory computer-readable storage medium of claim 49, wherein the computer-readable code, when executed by at least one processor, further causes the at least one processor to receive a user service request from the user terminal, and to retrieve the learned model based on the user service request.

51. The non-transitory computer-readable storage medium of claim 50, wherein the computer-readable code, when executed by at least one processor, further causes the at

least one processor to delete the learned model subsequent to providing the output data to the third party service.

Application No:	GB1614184.8	**Examiner:**	Dr Mark Lewney
Claims searched:	1-51	**Date of search:**	21 December 2016

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	1, 2, 4, 12, 13, 16, 17, 22-25, 35, 36, 38-40, 43, 44, 49, 50	US2015/294670 A1 (DOMINIK ET AL.) - See especially FIG. 1 and accompanying description.
X	1-4, 9, 12, 13, 15-19, 22-27, 32, 35-41, 43-46 & 49-51	US2011/285504 A1 (PUERTO ET AL.) - See especially figs. 2 & 3 and paragraphs [0049-50].
A	-	US2012/059655 A1 (CARTALES ET AL.) - See especially figs. 2-4 and paragraphs [0022-49].

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^{X} :

Worldwide search of patent documents classified in the following areas of the IPC

G10L; H04L; H04W

The following online and other databases have been used in the preparation of this search report

WPI, EPODOC.

International Classification:

Subclass	Subgroup	Valid From
H04L	0009/00	01/01/2006
G10L	0015/30	01/01/2013
G10L	0017/18	01/01/2013