(12) PATENT APPLICATION PUBLICATION
(19) INDIA
(22) Date of filing of Application :06/08/2020

(21) Application No.202041033610 A
(43) Publication Date : 11/02/2022

(54) Title of the invention : A METHOD OF TRAINING A SUBMODULE AND PREVENTING CAPTURE OF AN AI MODULE.

| | |
|---|---|
| (51) International classification | :G06N0020000000, H04L0029060000, G06N0003080000, G06F0003048800, G06N0003040000 |
| (31) Priority Document No | :NA |
| (32) Priority Date | :NA |
| (33) Name of priority country | :NA |
| (86) International Application No<br>Filing Date | :NA<br>:NA |
| (87) International Publication No | : NA |
| (61) Patent of Addition to Application Number<br>Filing Date | :NA<br>:NA |
| (62) Divisional to Application Number<br>Filing Date | :NA<br>:NA |

(71)**Name of Applicant :**
**1)Robert Bosch Engineering and Business Solutions Private Limited**
Address of Applicant :123, Industrial Layout, Hosur Road, Koramangala, Bangalore – 560095, Karnataka, India Karnataka India
**2)Robert Bosch GmbH**
(72)**Name of Inventor :**
**1)Parmar Manojkumar Somabhai**
**2)Yash Mayurbhai Thesia**

(57) Abstract :
The present disclosure proposes a method of training a submodule (14) and preventing capture of an AI module (12). Input data 5 is received from at least one user through an input interface (11). It is transmitted through a blocker module (18) to an AI module (12), which computes a first output data by executing a first model (M) based on the input data. A submodule (14) in the AI system (10) trained using methods steps (200) processes the 10 input data to identify an attack vector from the input data. The submodule (14) executes at least two models, where one model is the first model. The identification information of the attack vector is sent to the information gain module (16). 15 Figure 1.

No. of Pages : 21 No. of Claims : 6

**Applicants:** Robert Bosch Engineering and Business Solution Private Limited;
Robert Bosch GmbH.

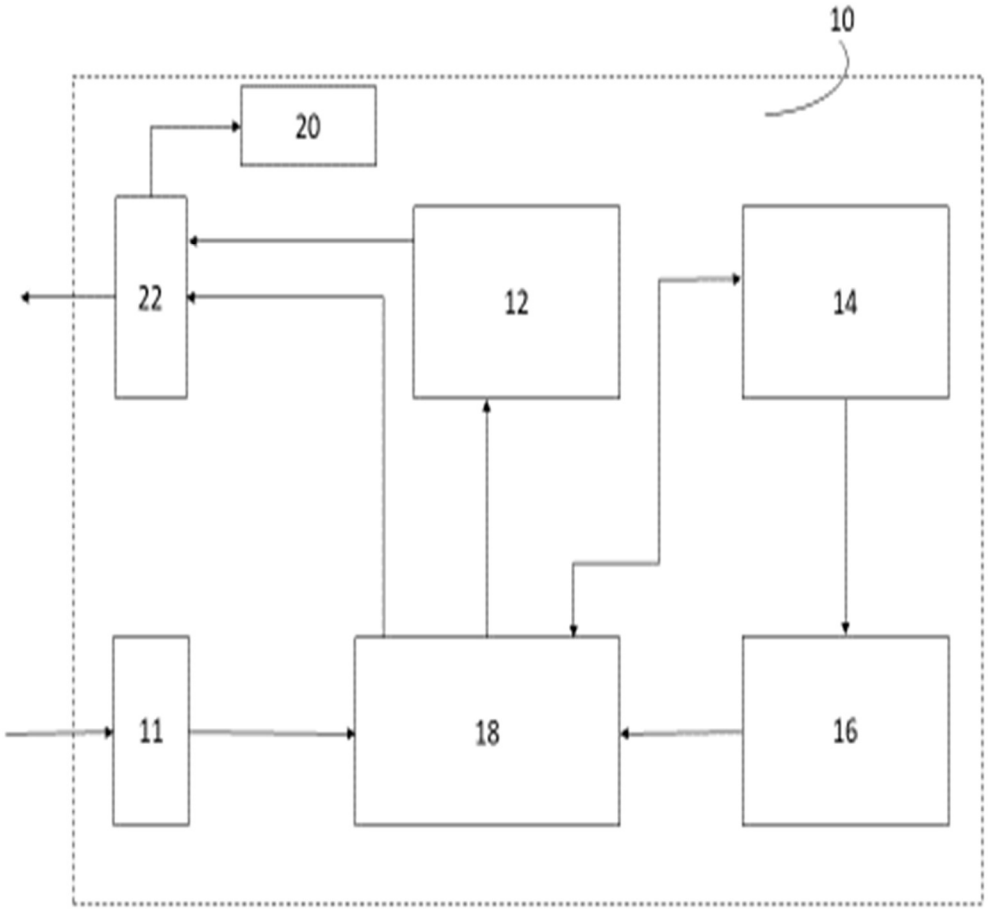



Figure 1

(Digitally signed)

Siddharth Karkhanis (IN/PA- 1195)

On behalf of the applicants.

**Applicants:** Robert Bosch Engineering and Business Solution Private Limited;
Robert Bosch GmbH.

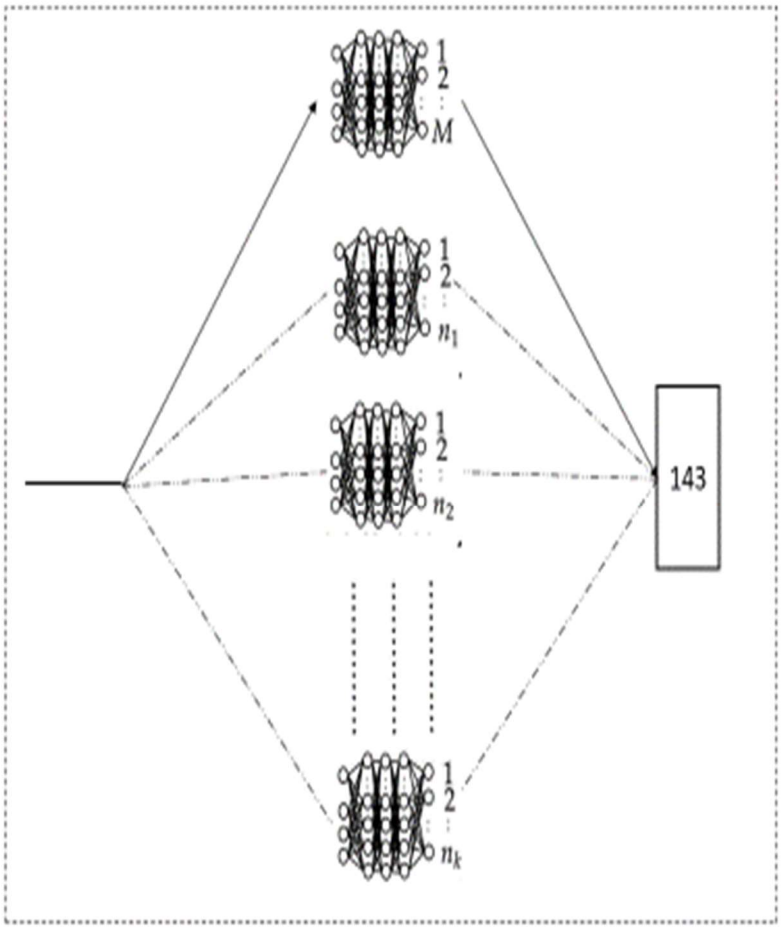



Figure 2

(Digitally signed)

Siddharth Karkhanis (IN/PA- 1195)

On behalf of the applicants.

**Applicants:** Robert Bosch Engineering and Business Solution Private Limited;
Robert Bosch GmbH.

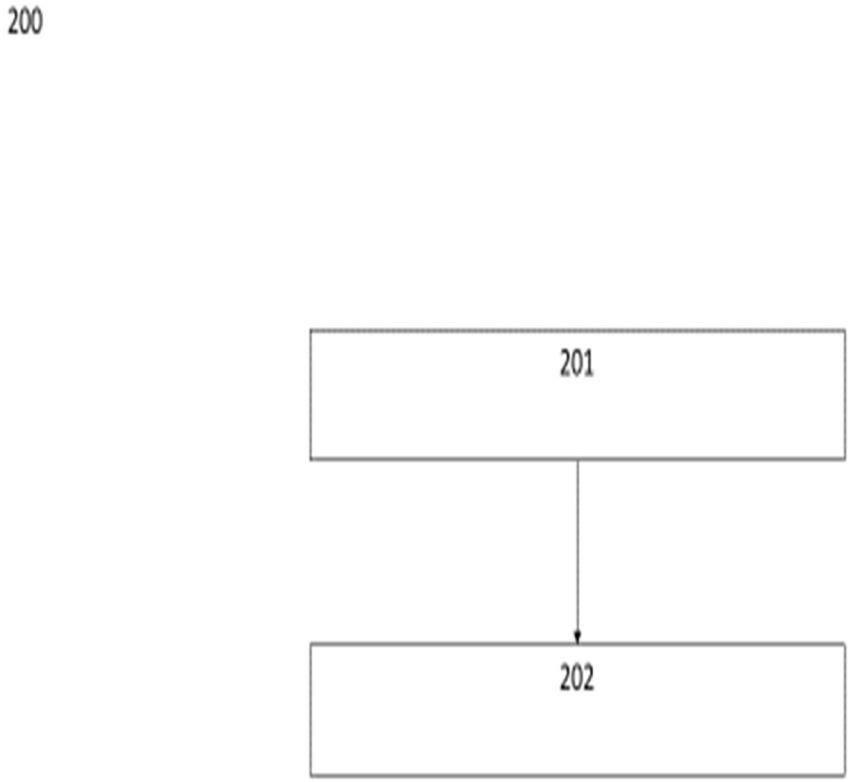



Figure 3

(Digitally signed)

Siddharth Karkhanis (IN/PA- 1195)

On behalf of the applicants.

**Applicants:** Robert Bosch Engineering and Business Solution Private Limited;

Robert Bosch GmbH.

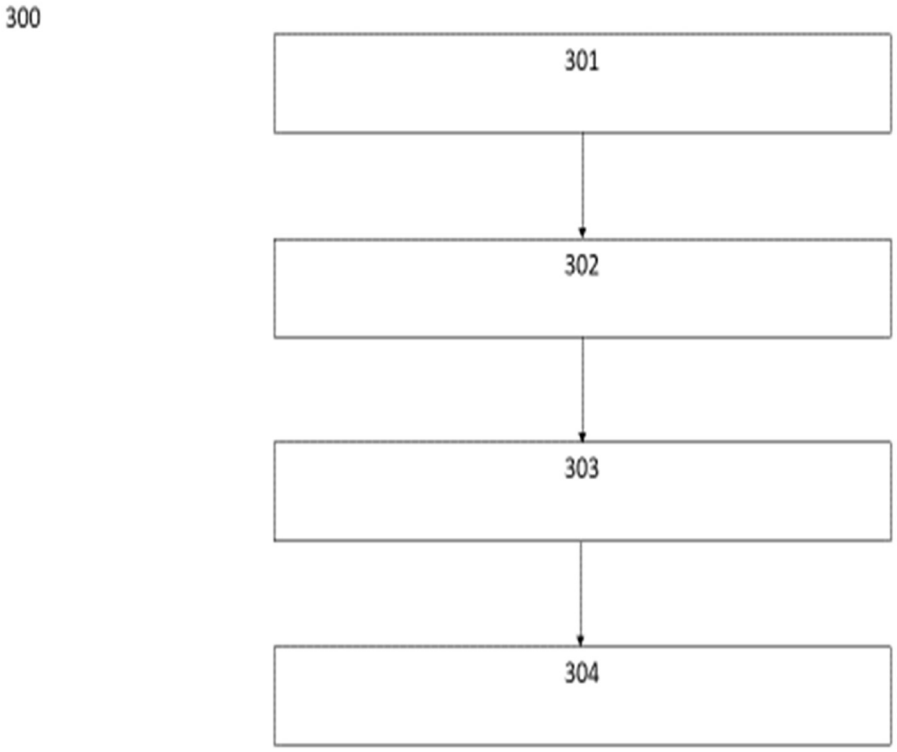



Figure 4

(Digitally signed)

Siddharth Karkhanis (IN/PA- 1195)

On behalf of the applicants.

**FORM 2**

**THE PATENTS ACT, 1970**

**(39 of 1970) & the Patents Rules 2003**

**COMPLETE SPECIFICATION**

**(SECTION 10 and Rule 13)**

**1. Title of the Invention:**

A method of training a submodule and preventing capture of an AI module.

**2. Applicants:**

a. Name: Robert Bosch Engineering and Business Solutions Private Limited

Nationality: INDIA

Address: 123, Industrial Layout, Hosur Road, Koramangala, Bangalore – 560095, Karnataka, India

b. Name: Robert Bosch GmbH

Nationality: GERMANY

Address: Feuerbach, Stuttgart, Germany

Complete Specification:

The following specification describes and ascertains the nature of this invention and the manner in which it is to be performed

**Field of the invention**

[0001] The present disclosure relates to a method of training a sub-module in an AI system and a method of preventing capture of an AI module in the AI system.

**Background of the invention**

[0002] With the advent of data science, data processing and decision making systems are implemented using artificial intelligence modules. The artificial intelligence modules use different techniques like machine learning, neural networks, deep learning etc. Most of the AI based systems, receive large amounts of data and process the data to train AI models. Trained AI models generate output based on the use cases requested by the user. Typically the AI systems are used in the fields of computer vision, speech recognition, natural language processing, audio recognition, healthcare, autonomous driving, manufacturing, robotics etc. where they process data to generate required output based on certain rules/intelligence acquired through training.

[0003] To process the inputs and give a desired output, the AI systems use various models/algorithms which are trained using the training data. Once the AI system is trained using the training data, the AI systems use the models to analyze the real time data and generate appropriate result. The models may be fine-tuned in real-time based on

the results. The models in the AI systems form the core of the system. Lots of effort, resources (tangible and intangible), and knowledge goes into developing these models.

[0004] It is possible that some adversary may try to capture/copy/extract the model from AI systems. The adversary may use different techniques to capture the model from the AI systems. One of the simple techniques used by the adversaries is where the adversary sends different queries to the AI system iteratively, using its own test data. The test data may be designed in a way to extract internal information about the working of the models in the AI system. The adversary uses the generated results to train its own models. By doing these steps iteratively, it is possible to capture the internals of the model and a parallel model can be built using similar logic. This will cause hardships to the original developer of the AI systems. The hardships may be in the form of business disadvantages, loss of confidential information, loss of lead time spent in development, loss of intellectual properties, loss of future revenues etc.

[0005] There are methods known in the prior arts to identify such attacks by the adversaries and to protect the models used in the AI system. The prior art US 20190095629A1- Protecting Cognitive Systems from Model Stealing Attacks discloses one such method. It discloses a method wherein the input data is processed by applying a trained model to the input data to generate an output vector having values for each of the

plurality of pre-defined classes. A query engine modifies the output vector by inserting a query in a function associated with generating the output vector, to thereby generate a modified output vector. The modified output vector is then output. The query engine modifies one or more values to disguise the trained configuration of the trained model logic while maintaining accuracy of classification of the input data.

**Brief description of the accompanying drawings**

[0006] An embodiment of the invention is described with reference to the following accompanying drawings:

[0007] Figure 1 depicts an AI system.

[0008] Figure 2 depicts a submodule in an AI system.

[0009] Figure 3 illustrates method steps of training a submodule in an AI system.

[0010] Figure 4 illustrates method steps to prevent capturing of an AI module in an AI system.

**Detailed description of the drawings**

[0011] It is important to understand some aspects of artificial intelligence (AI) technology and artificial intelligence (AI) based systems or artificial intelligence (AI) system. This disclosure covers two aspects of AI systems. The first aspect is related to the training of a submodule in the AI system and second aspect is related to the prevention of capturing of the AI module in an AI system.

[0012] Some important aspects of the AI technology and AI systems can be explained as follows. Depending on the architecture of the implements AI systems may include many components. One such component is an AI module. An AI module with reference to this disclosure can be explained as a component which runs a model. A model can be defined as reference or an inference set of data, which is use different forms of correlation matrices. Using these models and the data from these models, correlations can be established between different types of data to arrive at some logical understanding of the data. A person skilled in the art would be aware of the different types of AI models such as linear regression, naïve bayes classifier, support vector machine, neural networks and the like. It must be understood that this disclosure is not specific to the type of model being executed in the AI module and can be applied to any AI module irrespective of the AI model being executed. A person skilled in the art will also appreciate that the AI module may be implemented as a set of software instructions, combination of software and hardware or any combination of the same.

[0013] Some of the typical tasks performed by AI systems are classification, clustering, regression etc. Majority of classification tasks depend upon labeled datasets; that is, the data sets are labelled manually in order for a neural network to learn the correlation between labels and data. This is known as supervised learning. Some of the typical

applications of classifications are: face recognition, object identification, gesture recognition, voice recognition etc. Clustering or grouping is the detection of similarities in the inputs. The cluster learning techniques do not require labels to detect similarities. Learning without labels is called unsupervised learning. Unlabeled data is the majority of data in the world. One law of machine learning is: the more data an algorithm can train on, the more accurate it will be. Therefore, unsupervised learning models/algorithms has the potential to produce accurate models as training dataset size grows.

[0014] As the AI module forms the core of the AI system, the module needs to be protected against attacks. Attackers attempt to attack the model within the AI module and steal information from the AI module. The attack is initiated through an attack vector. In the computing technology a vector may be defined as a method in which a malicious code/virus data uses to propagate itself such as to infect a computer, a computer system or a computer network. Similarly an attack vector is defined a path or means by which a hacker can gain access to a computer or a network in order to deliver a payload or a malicious outcome. A model stealing attack uses a kind of attack vector that can make a digital twin/replica/copy of an AI module.

[0015] The attacker typically generates random queries of the size and shape of the input specifications and starts querying the model with

these arbitrary queries. This querying produces input-output pairs for random queries and generates a secondary dataset that is inferred from the pre-trained model. The attacker then take this I/O pairs and trains the new model from scratch using this secondary dataset. This is black box model attack vector where no prior knowledge of original model is required. As the prior information regarding model is available and increasing, attacker moves towards more intelligent attacks. The attacker chooses relevant dataset at his disposal to extract model more efficiently. This is domain intelligence model based attack vector. With these approaches, it is possible to demonstrate model stealing attack across different models and datasets.

[0016] It must be understood that the disclosure in particular discloses methodology used for training a submodule in an AI system and a methodology to prevent capturing of an AI module in an AI system. While these methodologies describes only a series of steps to accomplish the objectives, these methodologies are implemented in AI system, which may be a combination of hardware, software and a combination thereof.

[0017] Figure 1 depicts an AI system (10). The AI system (10) comprises an input interface (11), a blocker module (18), an AI module (12), a submodule (14), a blocker notification module (20), an information gain module (16) and at least an output interface (22). The input interface (11) receives input data from at least one user. The input interface (11) is

a hardware interface wherein a used can enter his query for the AI module (12).

[0018] The blocker module (18) is configured to block a user when the information gain calculated based on input attack queries exceeds a predefined threshold value. The blocker module (18) is further configured to modify a first output generated by an AI module (12). This is done only when the input is identified as an attack vector.

[0019] The AI module (12) to process said input data and generate the first output data corresponding to said input. The AI module (12) executes a model based on the input to generate a first output. As mentioned above this model could be any from the group of linear regression, naïve Bayes classifier, support vector machine, neural networks and the like.

[0020] The submodule (14) configured to identify an attack vector from the received input data. Figure 2 depicts the submodule (14) in an AI system (10). The submodule (14) comprises at least two models and a comparator (143). There are at least two models inside the submodule (14) that process the input. These at least two or more models again could be any from the group of linear regression, naïve Bayes classifier, support vector machine, neural networks and the like. However at least one of the models is the same as the one executed by the AI module (12). For example if the AI module (12) executes a convolutional neural network

(CNN) model, at least one module inside the submodule (14) will also execute the CNN model. A person skilled in the art will appreciate that similarly for other forms of data "n" number of models will be needed. The value of "n" is dynamic i.e. the no. of models executed by the submodule changes. This is dependent upon a current and historical values of information gain calculated by the information gain module. The comparator (143) receives and compares the output received on the execution of the various models with the same input.

[0021] The blocker notification module (20) transmits a notification to the owner of said AI system (10) on detecting an attack vector. The notification could be transmitted in any audio/visual/textual form.

[0022] The information gain module (16) is configured to calculate an information gain and send the information gain value to the blocker module (18). The information gain is calculated using the information gain methodology. In one embodiment, if the information gain extracted exceeds a pre-defined threshold, the AI system (10) is configured to lock out the user from the system. The locking out the system is initiated if the cumulative information gain extracted by plurality of users exceeds a pre-defined threshold.

[0023] The output interface (22) sends output to said at least one user. The output sent by the output interface (22) comprises the first output

data when the submodule (14) doesn't identify an attack vector from the received input. The output sent by the output interface (22) comprises a modified output received from the blocker module (18), when an attack vector is detected from the input.

[0024] It must be understood that each of the building blocks of the AI system (10) may be implemented in different architectural frameworks depending on the applications. In one embodiment of the architectural framework all the building block of the AI system (10) are implemented in hardware i.e. each building block may be hardcoded onto a microprocessor chip. This is particularly possible when the building blocks are physically distributed over a network, where each building block is on individual computer system across the network. In another embodiment of the architectural framework of the AI system (10) are implemented as a combination of hardware and software i.e. some building blocks are hardcoded onto a microprocessor chip while other building block are implemented in a software which may either reside in a microprocessor chip or on the cloud.

[0025] Figure 3 illustrates method steps (200) of training a submodule (14) in an AI system (10). The AI system (10) comprises the components described above in Figure 1 and 2. The submodule (14) is trained using a dataset used to train the AI module (12). The submodule (14) is trained using a dataset used to train the AI module (12). The

submodule (14) executes at least two models, said submodule (14) comprises a comparator for comparing output of at least two models. One of the models in the said at least two models is the first model (M). This first model (M) as explained in the preceding paragraphs is executed by the AI module (12).

[0026] In step 201, said at least two models receive the original dataset as input and are executed with the said input. The “n” models (N>1) contains the different classes for labels or number of classes. When the attack vector passes through all of these models, then class values across models may be different. If the class values across models are different then we consider the data pointer as attack vector. In step 202, the behavior of said submodule (14) is recorded.

[0027] Figure 4 illustrates method steps (300) to prevent capturing of an AI module (12) in an AI system (10). The AI system (10) and its components have been explained in the preceding paragraphs by means of figures 1 and 2. A person skilled in the art will understand that the submodule (14) trained by the method steps (200) is now used in real time for preventing capture of an AI module (12) in an AI system (10).

[0028] In method step 301, input interface (11) receives input data from at least one user. In step 302, this input data is transmitted through a blocker module (18) to an AI module (12). In step 303, the AI module

(12) computes a first output data by the AI module (12) executing a first model (M) based on the input data.

[0029] In step 304, in processed by submodule (14) to identify an attack vector from the input data, the identification information of the attack vector is sent to the information gain module (16). Processing the input data and said at least one subset of the input data further comprises two stages. First at least two models inside the submodule (14) are executed with the input data and said at least one subset. One of the models is the first model (M) i.e. it is same as the one executed by the AI module (12). Next the outputs received on execution of said at least two models is compared. An attack vector is determined from the input based on the comparison. If the outputs received are same, it means that's the input was not an attack vector. However if the comparator (143) finds difference in the outputs it inferred that the input is an attack vector.

[0030] Once the attack vector identification information is sent to the information gain module (16), an information gain is calculated. The information gain is sent to the blocker module (18). In an embodiment, if the information gain exceeds a pre-defined threshold, the user is blocked and the notification is sent the owner of the AI system (10) using blocker notification module (20). If the information gain is below a pre-defined threshold, although an attack vector was detected, the blocker module (18) may modify the first output generated by the AI module (12) to send it to the output interface (22).

[0031] In addition the user profile may be used to determine whether the user is habitual attacker or was it one time attack or was it only incidental attack etc. Depending upon the user profile, the steps for unlocking of the system may be determined. If it was first time attacker, the user may be locked out temporarily. If the attacker is habitual attacker then a stricter locking steps may be suggested.

[0032] It must be understood that the embodiments explained in the above detailed description are only illustrative and do not limit the scope of this invention. Any modification to a method of training a submodule (14) and preventing capture of an AI module (12) are envisaged and form a part of this invention. The scope of this invention is limited only by the claims.

**We Claim:**

1. An AI system (10) comprising at least:

an input interface (11) to receive input from at least one user;

an blocker module (18) configured to block at least one user, the blocker module (18) further configured to modify a first output generated by an AI module (12);

the AI module (12) to process said input data and generate first output data corresponding to said input;

a submodule (14) configured to identify an attack vector from the received input;

an information gain module (16) configured to calculate an information gain and send the information gain value to the blocker module (18);

a blocker notification module (20) to transmit a notification to the owner of said AI system (10) on detecting an attack vector; and

an output interface (22) to send an output to said at least one user.

2. The AI system (10) as claimed in claim 1, where the output sent by the output interface (22) comprises the first output data when the submodule (14) doesn't identify an attack vector from the received input.

3. A method of training a submodule (14) in an AI system (10), said AI system (10) comprising at least an AI module (12) executing a first model (M), a dataset used to train the AI module (12), said submodule

(14) executing at least two models, said submodule comprising a comparator to compare the output of at least two models, said method comprising the following steps:

executing at least two models in the submodule (14) with the dataset;

recording behavior of said submodule (14) .

4. The method of training a submodule (14) in an AI system (10) as claimed in claim 3, where one of the said at least two models is the first model (M).

5. A method to prevent capturing of an AI module (12) in an AI system (10), said method comprising the following steps:

receiving input data from at least one user through an input interface (11);

transmitting input data through a blocker module (18) to an AI module (12) ;

computing a first output data by the AI module (12) executing a first model (M) based on the input data;

processing input data by a submodule (14) to identify an attack vector from the input data, the identification information of the attack vector is sent to the information gain module (16).

6. The method to prevent capturing of an AI module (12) in an AI system (10) as claimed in claim 5, where processing the input data further comprises:

executing at least two models with the input data, one of the model is the first model (M);

comparing the outputs received on execution of said at least two models;

determining the input data as an attack vector based on the said comparison.

Dated this 30th day of july--------, 2020

(Digitally signed)

Siddharth Karkhanis (IN/PA- 1195)

On behalf of the applicants.

Formatted: Superscript

**TITLE: A method of training a submodule (14) and preventing capture of an AI module (12).**

**Abstract**

The present disclosure proposes a method of training a submodule (14) and preventing capture of an AI module (12). Input data is received from at least one user through an input interface (11). It is transmitted through a blocker module (18) to an AI module (12), which computes a first output data by executing a first model (M) based on the input data. A submodule (14) in the AI system (10) trained using methods steps (200) processes the input data to identify an attack vector from the input data. The submodule (14) executes at least two models, where one model is the first model. The identification information of the attack vector is sent to the information gain module (16).

Figure 1.