

METHODS AND SYSTEMS FOR IDENTITY PROOFING

FIELD OF INVENTION

[001] The present specification relates broadly, but not exclusively, to methods and systems for identity proofing.

BACKGROUND

[002] While conducting business, financial institutions need to meet regulatory requirements such as Anti-money laundering (AML) regulations. Particularly, financial-related businesses carried out by e-wallets are highly regulated globally. In order to avoid financial risks such as money laundering and fraud, regulators in many regions may require a natural person to provide a government-recognized identity document (ID) picture to prove their identity when opening an e-wallet account.

[003] At present, user information such as ID cards and pictures is mainly collected through online channels, such as web pages and mobile phone applications. For the operator of the e-wallets, during the user's account opening process, the uploaded ID picture needs to be checked to ensure the authenticity of the ID and to prevent fraudsters from using a fake ID to access the system. The process of verifying a user's identity is known as Know Your Customer (KYC). In some countries, the government has an official database and an open query interface. Merchants can connect to the official data sources for verification of ID information. However, such official database is not available in every country.

[004] Generally, ID cards issued by governments usually have some security features, such as light spots, watermarks, etc.. There are existing methods utilizing computer vision (CV) techniques for detecting fake IDs, which involve learning the security features of the ID cards and developing CV algorithms to detect if the uploaded ID picture is authentic or fake based on if the ID picture has these security features. However, nowadays fraudsters are able to forge ID pictures so well that the material, security features, and layout of the content on ID cards are highly similar to the real ID cards. As a result, these fake IDs can be

difficult to distinguish even for human eyes, and CV algorithms become insufficient to verify the authenticity of the IDs and detect fake IDs.

SUMMARY

[005] Described embodiments provide methods and systems for identity proofing by detecting fake identity documents (IDs). In some embodiments, the method uses space segmentation techniques to determine the authenticity of a new identity received. An identity can contain different attributes and values of the different attributes, and the occurrence times of the values within a certain period of time can be calculated. Segmentation is performed for the occurrence times with different conditions (e.g., by choosing different attributes of an identity) until the new identity is isolated from the historical data of identities. The number of segmentation times indicates a degree of abnormality, based on which the new identity can be determined to be a normal identity or a fake identity. In some implementations, the occurrence times may be adjusted based on a text frequency dictionary before segmentation is performed. Advantageously, the adjustment accounts for the inherent difference of how frequent different texts or values appear in reality.

[006] In some implementations, data of the identities include several attributes, such as name, date of birth, ID number, weight, height, eye color, expiry date of the ID, and address. In some implementations, when an ID is received in a form of an image (e.g., uploading a picture of an ID via a webpage or a mobile phone application), optical character recognition (OCR) algorithms can be used to extract the text related to the attributes and values of the attributes. Further, the occurrence times in a period of time for the different values of attributes are calculated and represented as a vector, known as a velocity vector, and segmentation is carried out to segment the velocity vector of the new identity from the entire vector space of velocity vectors of the identities in the historical data. The period of time can be in a form of a sliding window which stretches back in time from the present. Advantageously, frequently appearing values within the past period of time (e.g., five minutes) can be captured and reflected by a velocity vector with high values, and the identity (or identities) with the frequently appearing values can be identified as abnormal, as a result of a small number of segmentation times required to isolate the high value velocity vector.

[007] According to one embodiment, there is provided a method for identity proofing. The method includes: training a space segmentation model associated with historical data of identities, wherein the historical data comprises attributes of the identities, values of the attributes, and time stamps related to activities of the identities; receiving data corresponding to a new identity, wherein the data comprises attributes of the new identity, values of each of the attributes, and time stamps related to activities of the new identity; calculating, for the new identity and the historical data of identities, occurrence times of the values of the attributes between a first time stamp and a second time stamp; segmenting the new identity from the historical data of identities using the space segmentation model based on the occurrence times, wherein the new identity is isolated after a number of segmentation times; and proofing authenticity of the new identity based on the number of segmentation times.

[008] In some implementations, the occurrence times of the values of the attributes between a first time stamp and a second time stamp may be represented as a vector. In some implementations, the occurrence times may be adjusted based on text frequency and between the first and the second time stamp may be in a form of a sliding time window. In some implementations, the method may further include determining if the data received includes data corresponding to an image of an identity document and extracting the attributes and the values of the attributes from the image. In some implementations, the historical data of identities may be updated by including the new identity.

[009] According to other embodiments, one or more of these general and specific embodiments may be implemented using an apparatus including a plurality of modules, a system, a method, or a computer-readable media, or any combination of devices, systems, methods, and computer-readable media. The foregoing and other described embodiments can each, optionally, include some, none or all of the following embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Embodiments and implementations are provided by way of example only, and will be better understood and readily apparent to one of ordinary skill in the art from the following written description, read in conjunction with the drawings, in which:

[0011] Figure 1 is a flow chart illustrating an example of a method for identity proofing, according to an embodiment.

[0012] Figure 2 is a flow diagram illustrating an example of an implementation of the method for identity proofing in Figure 1, according to an embodiment.

[0013] Figure 3A is a schematic of an example of segmentation of a normal point in a two-dimension plane, according to an embodiment.

[0014] Figure 3B is a schematic of an example of a segmentation of an anomaly point in a two-dimension plane, according to an embodiment.

[0015] Figures 4 is a schematic diagram of an example of modules of a system for identity proofing, according to an embodiment.

[0016] Figure 5 is a block diagram of an example of a computer system suitable for executing at least some steps of the example methods shown in Figures 1 and 2, according to an embodiment.

[0017] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been depicted to scale. For example, the dimensions of some of the elements in the illustrations, block diagrams or flowcharts may be exaggerated in respect to other elements to help to improve understanding of the present embodiments.

DETAILED DESCRIPTION

[0018] Embodiments will be described, by way of example only, with reference to the drawings. Like reference numerals and characters in the drawings refer to like elements or equivalents.

[0019] Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those

requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

[0020] Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as “receiving”, “obtaining”, “training”, “determining”, “segmenting”, “calculating”, “generating”, “detecting”, “indicating”, “converting”, “adding”, “adjusting”, “comparing”, “updating”, “extracting”, “representing”, “proofing”, “authenticating”, “outputting”, or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

[0021] The present specification also discloses apparatuses for performing the operations of the methods. Such apparatuses may be specially constructed for the required purposes, or may comprise a computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate. The structure of a computer suitable for executing the various methods / processes described herein will appear from the description below.

[0022] In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the specification contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the scope of the specification.

[0023] Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a computer effectively results in an apparatus that implements the steps of the preferred method.

[0024] The present specification may also be implemented as hardware modules. More particularly, in the hardware sense, a module is a functional hardware unit designed for use with other components or modules. For example, a module may be implemented using discrete electronic components, or it can form a portion of an entire electronic circuit such as an Application Specific Integrated Circuit (ASIC) or Field Programmable Gate Array (FPGA). Numerous other possibilities exist. Those skilled in the art will appreciate that the system can also be implemented as a combination of hardware and software modules.

[0025] Identity proofing is the act or process of verifying the authenticity of an identity and can be considered as a form of fraud detection or fake identity detection, in which legitimacy of users are verified and potential fraudsters may be detected before fraudulent acts are carried out. Effective identity proofing can enhance data security of systems by permitting only authenticated users to access its protected resources. Embodiments seek to provide methods and systems for identity proofing which detect fake ID information or images uploaded by fraudsters. Advantageously, financial risks such as money laundering and fraud can be effectively reduced or eliminated.

[0026] The techniques described in this specification produce one or more technical effects. In some embodiments, space segmentation techniques based on values of identity attributes and identity behaviors are used for fake ID detection and identity proofing, which are effective in detecting pictures of fake IDs that highly imitate the real IDs. In some embodiments, the frequency of occurrence in a period of time for the different values of identity attributes are calculated and converted to vectors for performing space segmentation. The

vectors may reflect a degree of abnormality of the ID, which advantageously assist in recognizing frequently appearing values within the period of time and determining the authenticity of the identity. In some embodiments, the period of time can be in a form of a sliding window which stretches back in time from the present, which may identify fake ID pictures frequently uploaded within the past period of time by fraudsters.

[0027] Figure 1 is a flow chart 100 illustrating a method for identity authentication including the following steps:

- 110: training a space segmentation model associated with historical data of identities, wherein the historical data comprises attributes of the identities, values of the attributes, and time stamps related to activities of the identities;
- 120: receiving data corresponding to a new identity, wherein the data comprises attributes of the new identity, values of each of the attributes, and time stamps related activities of the new identity;
- 125: calculating, for the new identity and the historical data of identities, occurrence times of the values of the attributes between a first time stamp and a second time stamp;
- 130: segmenting the new identity from the historical data of identities using the space segmentation model based on the occurrence times, wherein the new identity is isolated after a number of segmentation times;
- 140: proofing authenticity of the new identity based on the number of segmentation times; and
- 150: updating the historical data of identities by including the new identity.

[0028] At step 110, a space segmentation model associated with historical data of identities is trained. In some embodiments, the trained space segmentation model can be a tree model. The historical data of identities may contain attributes of the identities and values of the attributes. Attributes of the identities may include some or all of the following: name, date of birth, address, height, weight, country, identity document (ID) number, or the like. Values of the attributes may be in a form of text (e.g., if attribute is name) or numbers (e.g., if attribute is date of birth), depending on the attributes. The historical data of identities may further contain data related to identity behaviors, such as time stamps related to the user activities. These activities may include opening an account, registration, updating identity information, logging in, logging out, and identity proofing such as performing KYC. Such identity behavior data is important in identity proofing and detecting

fake IDs, because fraudsters' activities can be different from normal users (e.g., uploading identities with repeated information or performing multiple KYC attempts within a short period of time).

[0029] At step 120, data corresponding to a new identity is received. The data may include attributes of the new identity, values of the attributes of the new identity, and identity behavior data such as time stamps related to the activities of the new identity. In some embodiments, data corresponding to the new identity may be received as image data, such as a picture of an ID card or ID page. The picture may be uploaded via webpages or mobile applications. In some implementations, first, the picture of the ID card or ID page is checked to determine whether the picture includes an image of an ID. For example, if a picture of an animal is uploaded, the picture will be determined to be unsatisfactory and the process will be terminated. In some implementations, an ID classification algorithm may be used for determining if the image data is acceptable.

[0030] In some embodiments, once the image data is checked to contain an image of an ID and determined to be acceptable, algorithms may be used to extract the attributes of the new identity and the values of the attributes from the image data. In some implementations, the algorithms may include an OCR algorithm.

[0031] At step 125, occurrence times of the values of the attributes between a first time stamp and a second time stamp are calculated for the new identity and the historical data of identities. In implementations, the calculations can be done for values of all the attributes in the data or selected attributes. The first and second time stamps can be set according to the period of interest. In some embodiments, the period of interest may be in a form of a sliding window that stretches back in time from the present, for example, the past five minutes. As such, the second time stamp can be set to be the present, and the first time stamp can be set as five minutes before the present. In some implementations, occurrence times of values of the attributes for each identity between two time stamps may be calculated and represented as a vector. Advantageously, the vector may reflect on a degree of abnormality of the identity based on the how frequent the values of the attributes of the identity appear within a period of time.

[0032] At step 130, the new identity is segmented from the historical data of identities using the space segmentation model developed at step 110, and based on the occurrence times calculated at step 125. By inputting the new identity (e.g., a vector representing the occurrence times of the values of attributes in the new identity) into the algorithm of the space segmentation model and performing segmentation for different attributes, the new identity can be isolated after a number of segmentation times. At step 140, authenticity of the new identity can be proofed based on the number of segmentation times. Some examples of isolating the new identity by space segmentation will be described in more details with reference to Figures 3A and 3B.

[0033] Optionally, at step 150, the new identity can be added to the history data of identities and the history data can be updated to include the data received corresponding to the new identity. Advantageously, the history data can include the new identity, its related attributes and values of the attributes, and the behavior data of the new identity, which may enhance the future training models and improve the accuracy.

[0034] Figure 2 is a flow diagram 200 illustrating an implementation of the method for identity proofing as described above with reference to Figure 1. In the beginning of the process, a picture 210 of an ID card or ID page is received. An ID classification algorithm may be used to determine whether the picture 210 is an ID (i.e., not a photo of an animal or other unacceptable pictures). If the picture 210 is determined to be an ID card or ID page, continue with the process. If not, the process is terminated.

[0035] In the next step, an OCR algorithm may be used to extract texts related to personal information on the ID card or ID page, such as name, ID number, address, date of birth, etc. Such attributes of the identity are shown as “Field”, and the values of the attributes are shown as “Value”. Subsequently, frequency of occurrence of each value within a predetermined period of time (i.e., between two predetermined time stamps) may be calculated, and the occurrence times of the values may be generated as a vector, known as a velocity vector. In some embodiments, the velocity vector is constructed as: $D(\text{VelocityVector})=[V(\text{Name}), V(\text{DOB}), V(\text{Address}), V(\text{IDNumber}), \dots]$ by combining the velocity of each text item to form the velocity vector that is specific to this identity. As illustrated in Figure 2, within a predetermined period of time, if the name “James” occurred 3

times, the date of birth “8/1/1990” occurred 1 time, the address (or state) “Kansas” occurred 1 time, and the ID number “123456” occurred 7 times, the velocity vector generated will be [3, 1, 1, 7]. In some implementations, the predetermined period of time can be in a form of a sliding window, which is a stream mode that allows events to be matched over a time period that stretches back in time from the present. For example, a sliding window of two minutes includes all the events that have occurred in the past two minutes. Practically, identity information has a low possibility to repeat, and an identity can only be used once when registering an account. As such, if certain identity field values appear too frequently, they can be considered as anomaly. In real life, as fraudsters are likely to upload a batch of fake IDs within a certain period of time, implementation of the frequency of occurrence and sliding window logic may advantageously detect such fraudster activities.

[0036] After generating the velocity vector representing the new identity received in the ID card picture 310, the velocity vector is inputted into an algorithm based on space segmentation. The algorithm may cause segmentation to be performed with different conditions until the velocity vector is isolated from the entire vector space. The algorithm may output a number of segmentation times, which may indicate the degree of abnormality of the velocity vector corresponding to the new identity. In some implementations, the values in the velocity vector may be adjusted based on a text frequency dictionary. As the frequency of occurrence for different texts in reality is inherently different, this factor is advantageously taken into account by this adjustment, which makes the number of segmentation time in the output more reliable. For example, if a name appears frequently in reality, such as “Smith”, the corresponding value in the velocity vector is inherently higher than other names, which may result in the velocity vector being identified as an anomaly point. Such false outputs can be reduced if the text frequency dictionary is used to adjust the value in the velocity vector $V(\text{Name})$. Finally, the new identity received will be determined as normal or anomaly based on the number of segmentation times to isolate the adjusted velocity vector.

[0037] An example space segmentation algorithm or isolation algorithm is presented below. After generating a velocity vector for the new identity, the velocity vector is mapped to a point on a high-dimensional space by suitable methods, and then segmented using different hyperplanes (straight lines in a two-dimensional space) until a separate subspace contains only this point.

Algorithm: SpaceSegment(x, T, e)

Input: x – an instance, e – current number of segmentation times, T – a trained tree model

Output: y – number of segmentation times

1. *If x is isolated then*
2. *Return y*
3. *End if*
4. *$a = T.splitAttr$*
5. *If $x.a < T.a.splitValue$ then*
6. *Return SpaceSegment($x, T.left, e+1$)*
7. *Else $x.a \geq T.a.splitValue$ then*
8. *Return SpaceSegment($x, T.right, e+1$)*
9. *End if*

As shown in the algorithm, for a new input x corresponding to a new identity, the identity may include different attributes [attribute1, attribute2, attribute3...]. Similarly, there are same attributes inside the trained tree model T based on historical data of identities, and x will be segmented based on these attributes. For a tree model, the process of segmentation can be considered as the process of placing x on the top of tree T , and then falling to the bottom of the tree T layer by layer. For each split, an attribute is randomly selected from [attribute1, attribute2, attribute3...] as the *splitAttribute*, and a value is randomly selected as a *splitValue* in the value range of occurrence times for this attribute. The *splitValue* is compared with the attribute value in the velocity vector corresponding to x . According to the comparison result, x will fall into the left or right subtree of a layer, and then do the next segmentation until x falls to the leaf node, which means the segmentation process is complete. The number of segmentation times will be returned as output.

[0038] Figure 3A is a schematic 300 of segmentation of a normal point in a two-dimension plane, and Figure 3B is a schematic 350 of segmentation of an anomaly point in a two-dimension plane. As an example, the two-dimension plane represents occurrence times between two time stamps (i.e., velocity) of values for two selected attributes, Name and ID Number. Referring to Figure 3A, Point X_{normal} has a velocity vector value of [2, 1], which is among common values in this group, as most identities have a $V(Name)$ that is low and a

$V(\text{IDnumber})$ that is 1. As a result, the Point X_{normal} is mixed within many neighbour points having a similar value. When space segmentation is performed, a number of 11 segmentation times is needed to isolate Point X_{normal} . Referring to Figure 3B and using the previous example in Figure 2, Point X_{anomaly} has a velocity vector value of [3, 7], which is a relatively rare value in the group. As a result, the Point X_{anomaly} is far from the majority of points and only a number of 4 segmentation times is needed to isolate Point X_{anomaly} . One may appreciate that, while a two-dimension plane example is presented, the space segmentation techniques can be applied to a multi-dimension space and a point can be isolated or divided by multiple hyperplanes, where multiple attributes are selected.

[0039] From the above embodiments, one may appreciate that the method may transform a picture of an ID card or ID page into a velocity factor that reflects the degree of abnormality of the identity, and convert the problem of fake ID detection using computer vision into a space segmentation approach, which is a completely different perspective.

[0040] Figure 4 is a schematic diagram of an example system 400 including modules for identity proofing. The system 400 at least includes a processor module 402 and a memory module 404. The processor module 402 and the memory module 404 are interconnected. The memory module 404 stores history data 420 and computer program code (not shown in Figure 4). The memory module 404 and the computer program code are configured to, with the processor module 402, cause the system 400 to perform the steps for identity proofing as described in the present specification. The system 400 can include a receiver module 406, a training module 408, a generator module 410, a calculation module 412, an output module 416 and an update module 418. With reference to Figure 1 and Figure 2, the receiver module 406 can be configured to receive data corresponding to the identities. The training module 408 can be configured to train the space segmentation model, the ID classification model and the OCR model. The calculation module 412 can be configured to calculate occurrence times for values of the attributes of the identities within a predetermined period. The generator module 410 can be configured to generate vectors and vector space based on the calculated occurrence times. The output module 416 can be configured to output the number of segmentation time and/or the result if a new identity is determined to be a fraudulent identity. The update

module 418 can be configured to update the history data 420 by including the new identity received by the receiver module 406. One or more or any combination of these modules can be part of an apparatus for identity authentication.

[0041] The system, apparatus, module, or unit illustrated in the previous embodiments can be implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical embodiment device is a computer (and the computer can be a personal computer), a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an email receiving and sending device, a game console, a tablet computer, a wearable device, or any combination of these devices. The modules described as separate parts may or may not be physically separate, and parts displayed as modules may or may not be physical modules, may be located in one position, or may be distributed on a number of network modules. Some or all of the modules can be selected based on actual demands to achieve the objectives of the solutions of the specification. A person of ordinary skill in the art can understand and implement the embodiments of the present application without creative efforts.

[0042] Figure 5 is a block diagram of an example computer system 500 suitable for executing at least some steps of the example methods shown in Figures 1 and 2. The following description of the computer system / computing device 500 is provided by way of example only and is not intended to be limiting.

[0043] As shown in Fig. 5, the example computing device 500 includes a processor 502 for executing software routines. Although a single processor is shown for the sake of clarity, the computing device 500 may also include a multi-processor system. The processor 502 is connected to a communication infrastructure 506 for communication with other components of the computing device 500. The communication infrastructure 506 may include, for example, a communications bus, cross-bar, or network.

[0044] The computing device 500 further includes a main memory 504, such as a random access memory (RAM), and a secondary memory 510. The secondary memory 510 may include, for example, a storage drive 512, which may be a hard disk drive, a solid state drive or a hybrid drive and/or a removable storage drive 514, which may include a magnetic tape drive, an optical disk drive, a solid state storage

drive (such as a USB flash drive, a flash memory device, a solid state drive or a memory card), or the like. The removable storage drive 514 reads from and/or writes to a removable storage medium 518 in a well-known manner. The removable storage medium 518 may include magnetic tape, optical disk, non-volatile memory storage medium, or the like, which is read by and written to by removable storage drive 514. As will be appreciated by persons skilled in the relevant art(s), the removable storage medium 518 includes a computer readable storage medium having stored therein computer executable program code instructions and/or data.

[0045] In an alternative implementation, the secondary memory 510 may additionally or alternatively include other similar means for allowing computer programs or other instructions to be loaded into the computing device 500. Such means can include, for example, a removable storage unit 522 and an interface 520. Examples of a removable storage unit 522 and interface 520 include a program cartridge and cartridge interface (such as that found in video game console devices), a removable memory chip (such as an EPROM or PROM) and associated socket, a removable solid state storage drive (such as a USB flash drive, a flash memory device, a solid state drive or a memory card), and other removable storage units 522 and interfaces 520 which allow software and data to be transferred from the removable storage unit 522 to the computer system 500.

[0046] The computing device 500 also includes at least one communication interface 524. The communication interface 524 allows software and data to be transferred between computing device 500 and external devices via a communication path 526. In various embodiments of the specification, the communication interface 524 permits data to be transferred between the computing device 500 and a data communication network, such as a public data or private data communication network. The communication interface 524 may be used to exchange data between different computing devices 500 which such computing devices 500 form part an interconnected computer network. Examples of a communication interface 524 can include a modem, a network interface (such as an Ethernet card), a communication port (such as a serial, parallel, printer, GPIB, IEEE 1394, RJ45, USB), an antenna with associated circuitry and the like. The communication interface 524 may be wired or may be wireless. Software and data transferred via the communication interface 524 are in the form of signals which can be electronic, electromagnetic, optical or other signals capable of being received by communication interface 524. These signals are provided to the communication interface via the communication path 526.

[0047] As shown in Fig. 5, the computing device 500 further includes a display interface 528 which performs operations for rendering images to an associated display 530 and an audio interface 532 for performing operations for playing audio content via associated speaker(s) 534.

[0048] As used herein, the term "computer program product" may refer, in part, to removable storage medium 518, removable storage unit 522, a hard disk installed in storage drive 512, or a carrier wave carrying software over communication path 526 (wireless link or cable) to communication interface 524. Computer readable storage media refers to any non-transitory, non-volatile tangible storage medium that provides recorded instructions and/or data to the computing device 500 for execution and/or processing. Examples of such storage media include magnetic tape, CD-ROM, DVD, Blu-ray™ Disc, a hard disk drive, a ROM or integrated circuit, a solid state storage drive (such as a USB flash drive, a flash memory device, a solid state drive or a memory card), a hybrid drive, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computing device 500. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computing device 500 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

[0049] The computer programs (also called computer program code) are stored in main memory 504 and/or secondary memory 510. Computer programs can also be received via the communication interface 524. Such computer programs, when executed, enable the computing device 500 to perform one or more features of embodiments discussed herein. In various embodiments, the computer programs, when executed, enable the processor 607 to perform features of the above-described embodiments. Accordingly, such computer programs represent controllers of the computer system 500.

[0050] Software may be stored in a computer program product and loaded into the computing device 500 using the removable storage drive 514, the storage drive 512, or the interface 520. The computer program product may be a non-transitory computer readable medium. Alternatively, the computer program product may be

downloaded to the computer system 500 over the communication path 526. The software, when executed by the processor 502, causes the computing device 500 to perform the necessary operations to execute the method as shown in Figures 1 and 2.

[0051] It is to be understood that the embodiment of Fig. 5 is presented merely by way of example to explain the operation and structure of the system 500. Therefore, in some embodiments one or more features of the computing device 500 may be omitted. Also, in some embodiments, one or more features of the computing device 500 may be combined together. Additionally, in some embodiments, one or more features of the computing device 500 may be split into one or more component parts.

[0052] It will be appreciated that the elements illustrated in Fig. 5 function to provide means for performing the various functions and operations of the system as described in the above embodiments.

[0053] It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present specification as shown in the specific embodiments without departing from the scope of the specification as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

CLAIMS

What is claimed is:

1. A computer-implemented method for identity proofing, the method comprising:
training a space segmentation model associated with historical data of identities, wherein the historical data comprises attributes of the identities, values of the attributes of the identities, and time stamps related to activities of the identities;
receiving data corresponding to a new identity, wherein the data comprises attributes of the new identity, values of the attributes of the new identity, and time stamps related to activities of the new identity;
calculating, for the new identity and the historical data of identities, occurrence times of the values of the attributes between a first time stamp and a second time stamp;
segmenting the new identity from the historical data of identities using the space segmentation model based on the occurrence times, wherein the new identity is isolated after a number of segmentation times; and
proofing authenticity of the new identity based on the number of segmentation times.
2. The method as claimed in claim 1, wherein the occurrence times of the values of the attributes between a first time stamp and a second time stamp are represented as a vector.
3. The method as claimed in claim 1 or 2, wherein the occurrence times are adjusted based on text frequency.
4. The method as claimed in any one of the preceding claims, wherein between the first time stamp and the second time stamp is in a form of a sliding time window.
5. The method as claimed in any one of the preceding claims, further comprising:
determining if the data received comprises data corresponding to an image of an identity document; and
in response to the data received is determined to comprise data corresponding to an image of an identity document, extracting the attributes of the new identity and the values of the attributes of the new identity from the image.

6. The method as claimed in claim 5, wherein determining if the data received comprises data corresponding to an image of an identity document is based on a classification algorithm.
7. The method as claimed in claim 5 or 6, wherein extracting the attributes of the new identity and the values of the attributes of the new identity from the image is based on an optical character recognition (OCR) algorithm.
8. The method as claimed in any one of the preceding claims, wherein the trained space segmentation model comprises a tree model.
9. The method as claimed in any one of the preceding claims, the method further comprising:
 - updating the historical data of identities by including the new identity.
10. The method as claimed in any one of the preceding claims, wherein the attribute of the identities is selected from the group comprising name, date of birth, place of birth, height, weight, identity document number, and address.
11. The method as claimed in any one of the preceding claims, wherein the activity of the identities is selected from the group comprising registration, updating information, logging in, logging out, and identity proofing.
12. The method as claimed in claim 11, wherein the identity proofing comprises a Know Your Customer (KYC) process.
13. A system for identity authentication, the system comprising:
 - one or more processors; and
 - one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to perform the method of any one of claims 1 to 12.
14. An apparatus for identity authentication, the apparatus comprising a plurality of modules for performing the method of any one of claims 1 to 12.