US 20200134487A1

(19) **United States**

(12) **Patent Application Publication**

Kim et al.

(10) **Pub. No.: US 2020/0134487 A1**

(43) **Pub. Date: Apr. 30, 2020**

(54) **APPARATUS AND METHOD FOR PREPROCESSING SECURITY LOG**

(71) Applicant: **SAMSUNG SDS CO., LTD.**, Seoul (KR)

(72) Inventors: **Jang-Ho Kim**, Seoul (KR); **Young-Min Cho**, Seoul (KR); **Jung-Bae Jun**, Seoul (KR); **Seong-Hyeok Seo**, Seoul (KR); **Jang-Mi Shin**, Seoul (KR)

(21) Appl. No.: **16/665,663**

(22) Filed: **Oct. 28, 2019**

(30) **Foreign Application Priority Data**

Oct. 30, 2018 (KR) 10-2018-0130743

Publication Classification

(51) **Int. Cl.**

G06N 5/04	(2006.01)
G06F 16/25	(2006.01)
G06N 20/00	(2006.01)

(52) **U.S. Cl.**

CPC ***G06N 5/04*** (2013.01); ***G06N 20/00*** (2019.01); ***G06F 16/258*** (2019.01)

(57) **ABSTRACT**

According to one embodiment, An apparatus for preprocessing a security log includes a field divider configured to divide a character string of a security log into a plurality of fields on the basis of a structure of the security log, an ASCII code converter configured to convert a character string included in each of the plurality of divided fields into ASCII codes, and a vector data generator configured to generate vector data for each of the plurality of divided fields using the converted ASCII codes.

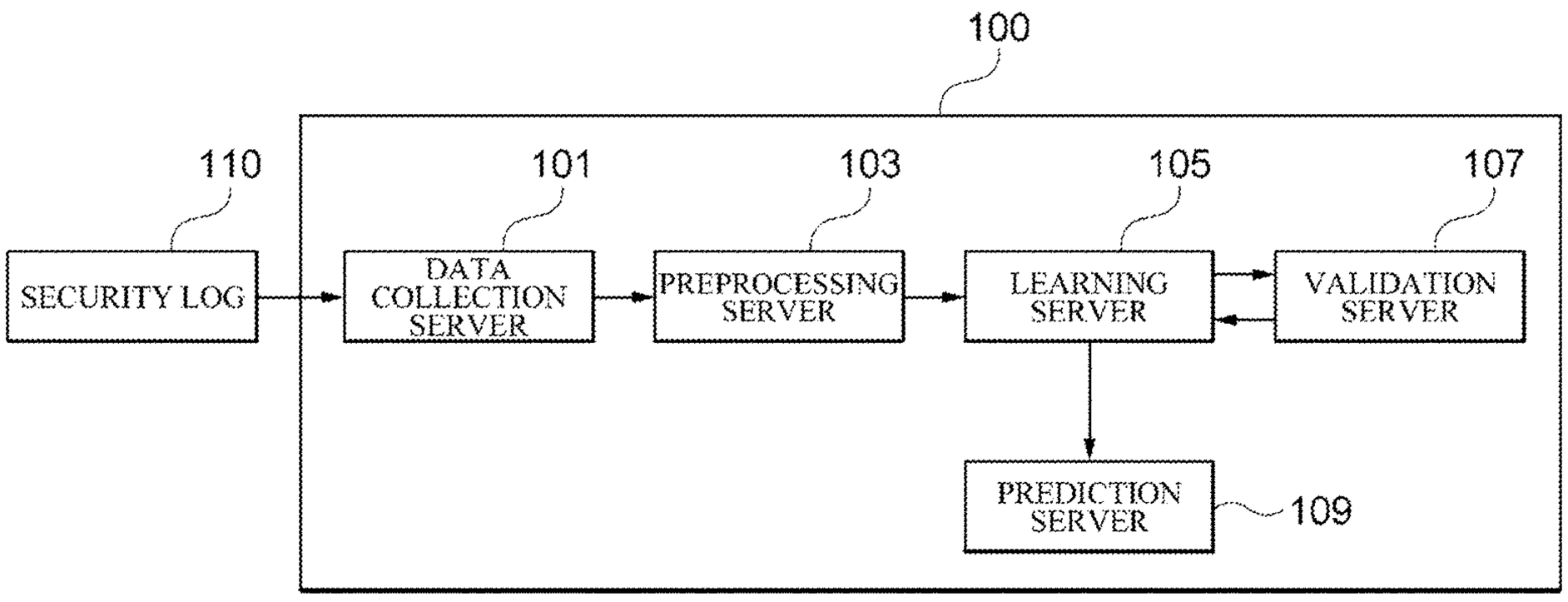


FIG. 1

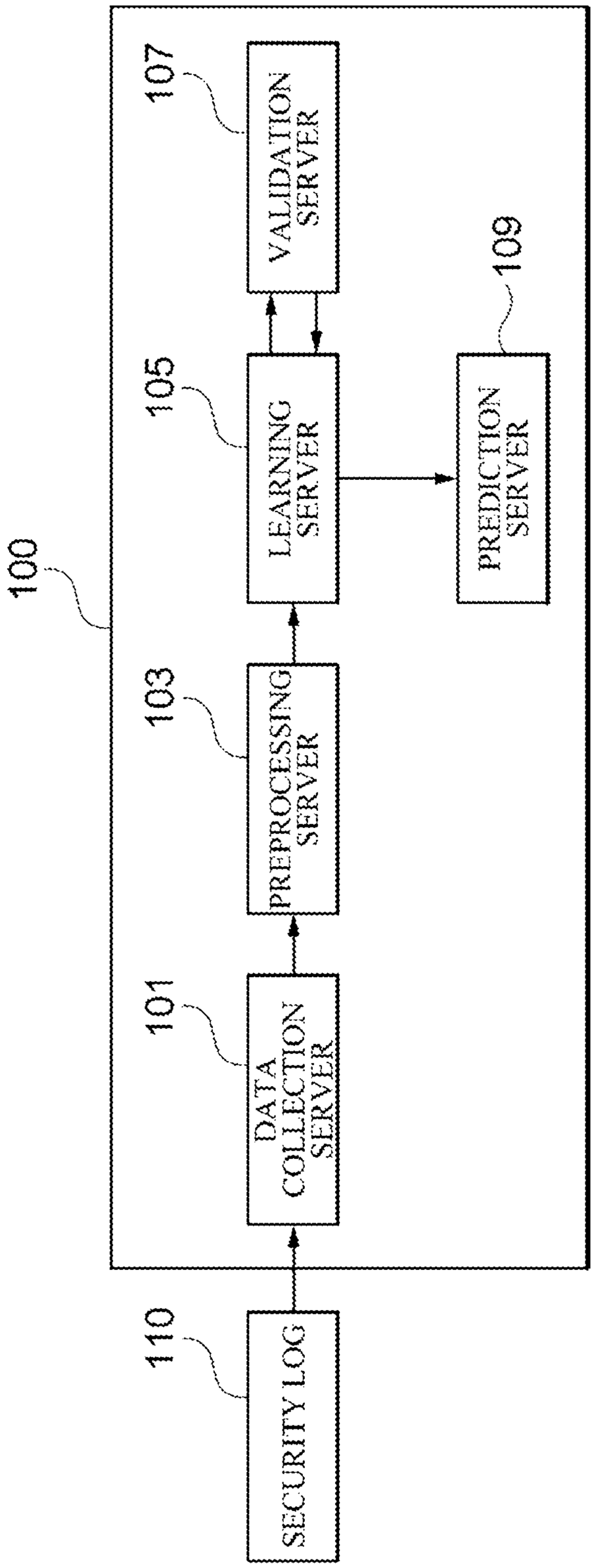

100
101
DATA COLLECTION SERVER
103
PREPROCESSING SERVER
105
LEARNING SERVER
107
VALIDATION SERVER
109
PREDICTION SERVER
110
SECURITY LOG

FIG. 2

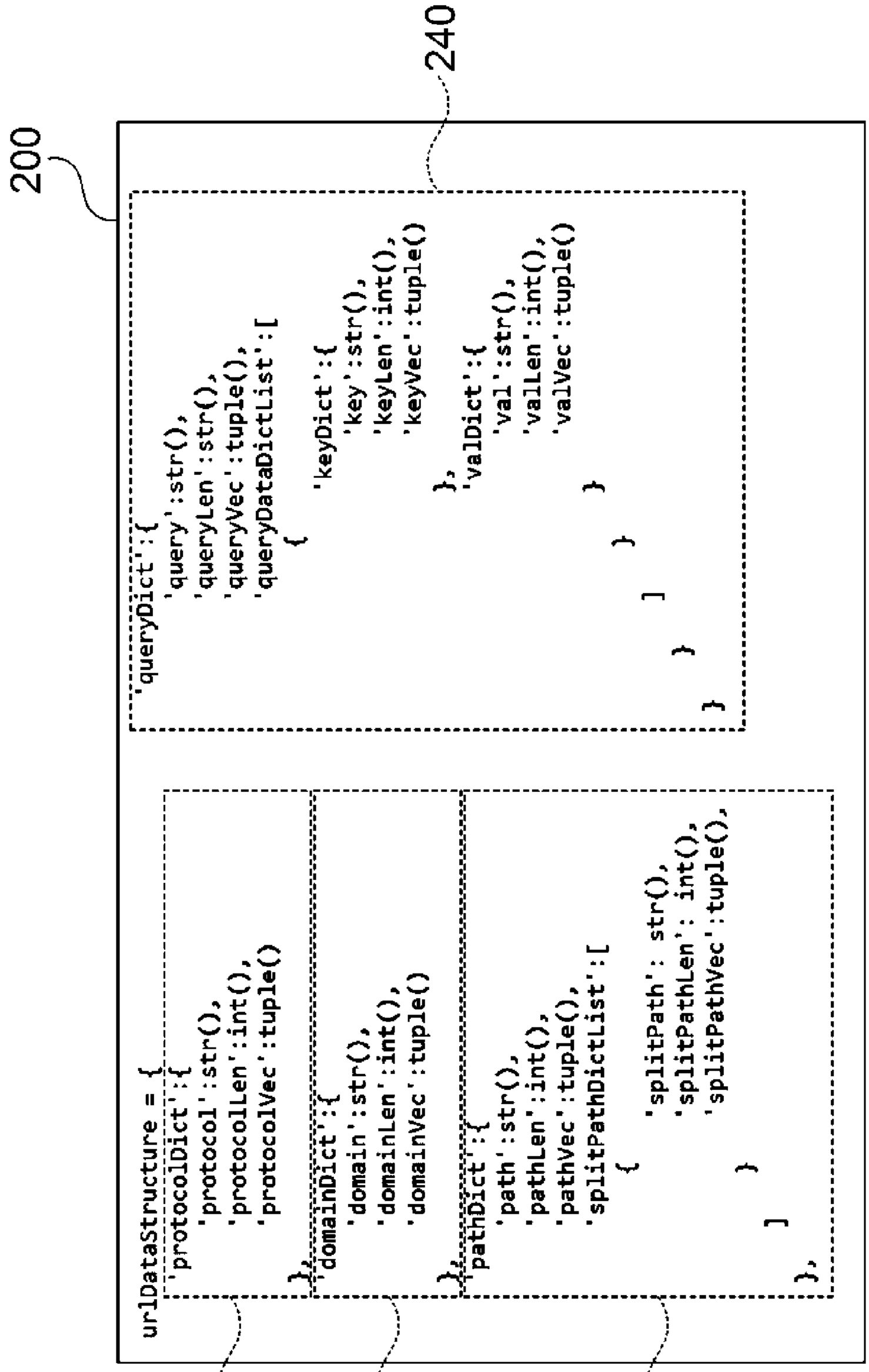

240
200
urlDataStructure = {
'protocolDict':{
'protocol':str(),
'protocolLen':int(),
'protocolVec':tuple()
},
'domainDict':{
'domain':str(),
'domainLen':int(),
'domainVec':tuple()
},
'pathDict':{
'path':str(),
'pathLen':int(),
'pathVec':tuple()
'splitPathDictList':[
{
'splitPath': str(),
'splitPathLen': int(),
'splitPathVec':tuple(),
}
]
},
'queryDict':{
'query':str(),
'queryLen':str(),
'queryVec':tuple()
'queryDataDictList':[
{
'keyDict':{
'key':str(),
'keyLen':int(),
'keyVec':tuple()
},
'valDict':{
'val':str(),
'valLen':int(),
'valVec':tuple()
}
}
]
}
210
220
230

FIG. 3

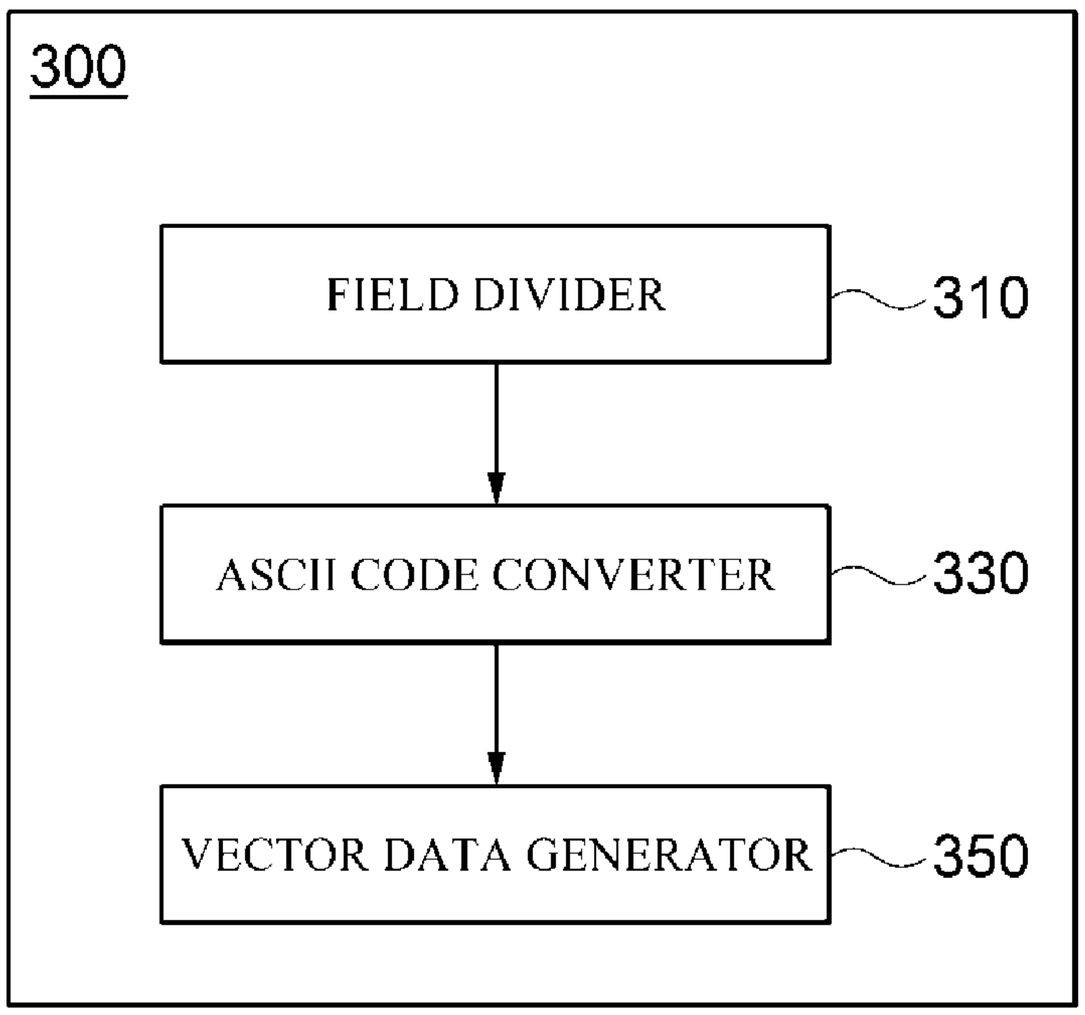

300
FIELD DIVIDER
310
ASCII CODE CONVERTER
330
VECTOR DATA GENERATOR
350

FIG. 4

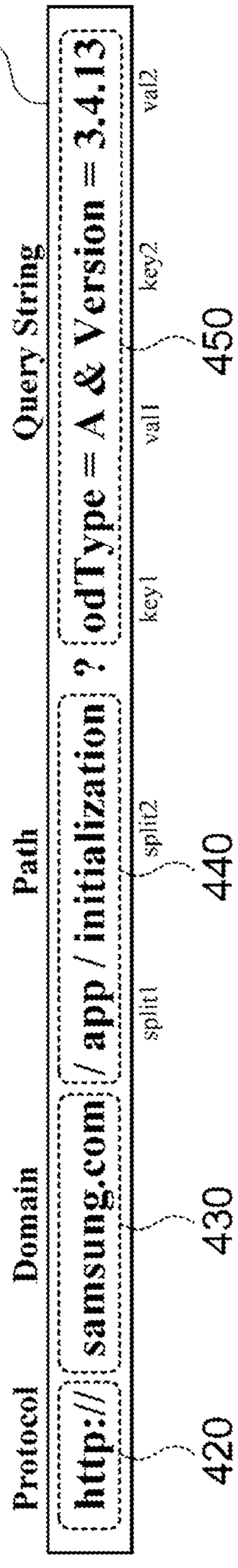

Protocol
Domain
Path
Query String
http://
samsung.com
/ app / initialization
? odType = A & Version = 3.4.13
split1
split2
key1
val1
key2
val2
410
420
430
440
450

FIG. 5

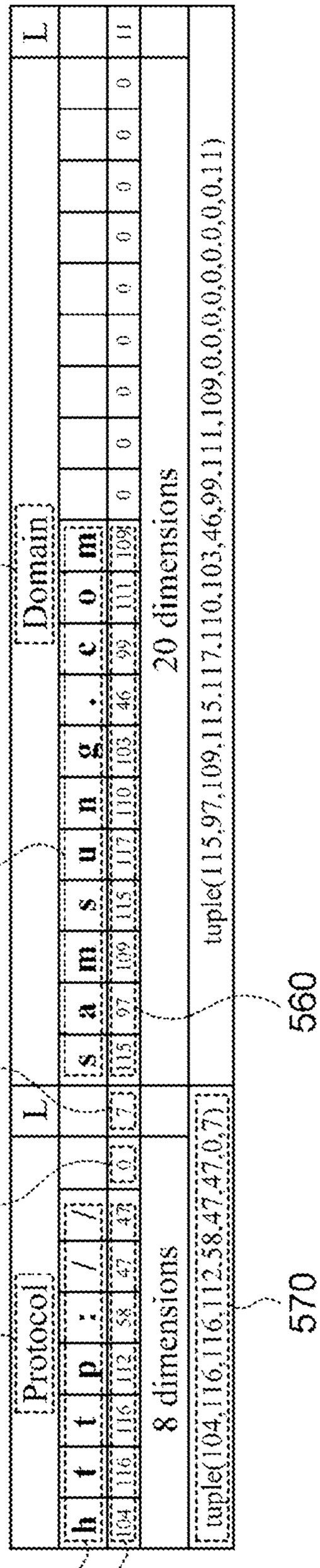

510
520
530
540
550
560
570
580
590
Protocol
Domain
8 dimensions
20 dimensions
tuple(104,116,116,112,58,47,47,0,7)
tuple(115,97,109,115,117,110,103,46,99,111,109,0,0,0,0,0,0,0,11)

FIG. 6

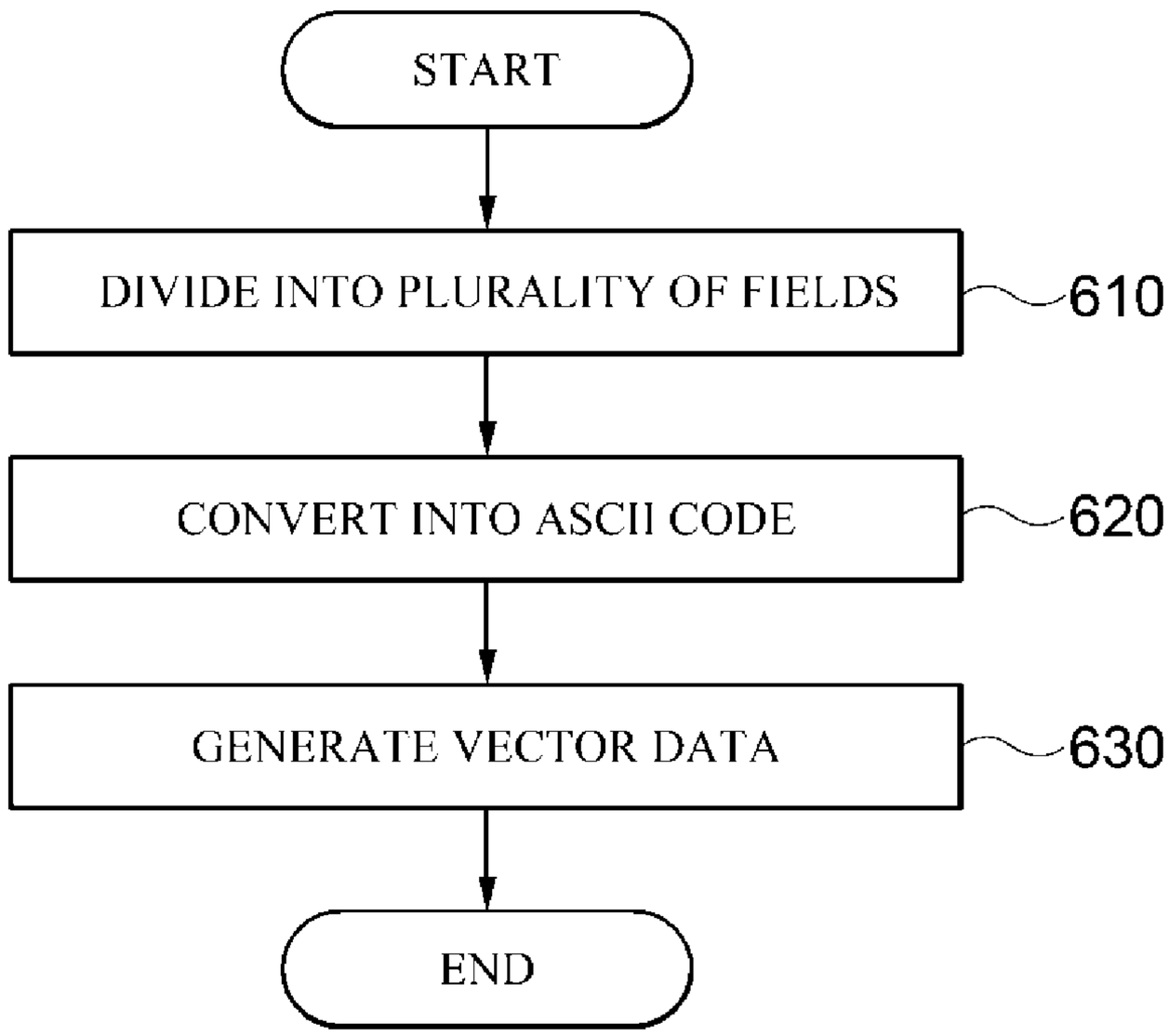
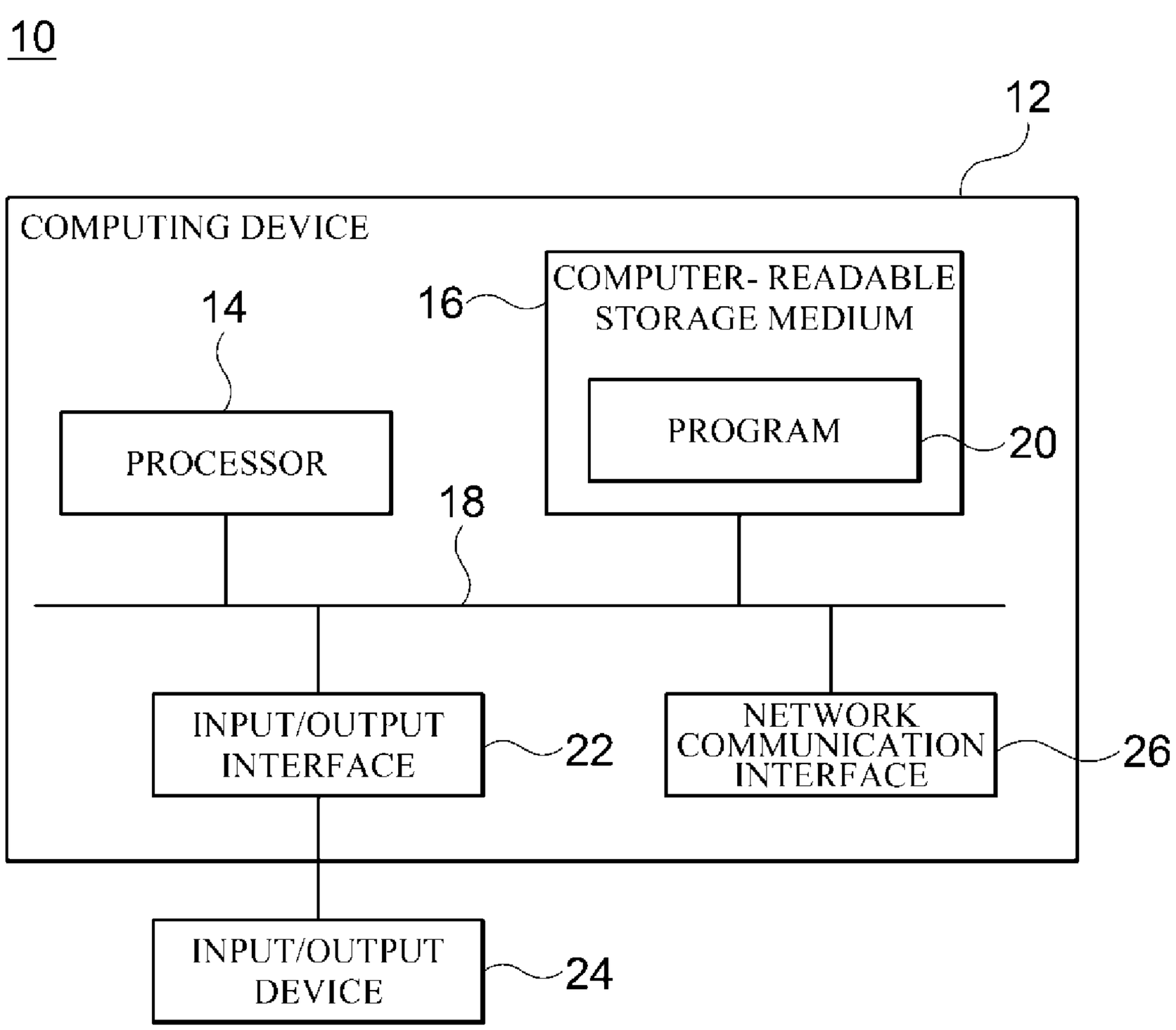

START
DIVIDE INTO PLURALITY OF FIELDS
610
CONVERT INTO ASCII CODE
620
GENERATE VECTOR DATA
630
END

FIG. 7


10
12
COMPUTING DEVICE
14
PROCESSOR
16
COMPUTER- READABLE STORAGE MEDIUM
PROGRAM
20
18
INPUT/OUTPUT INTERFACE
22
NETWORK COMMUNICATION INTERFACE
26
INPUT/OUTPUT DEVICE
24

APPARATUS AND METHOD FOR PREPROCESSING SECURITY LOG

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application claims the benefit under 35 USC § 119(a) of Korean Patent Application No. 10-2018-0130743, filed on Oct. 30, 2018, in the Korean Intellectual Property Office, the entire disclosure of which is incorporated herein by reference for all purposes.

BACKGROUND

1. Field

[0002] The following description relates to a technology for preprocessing a security log.

2. Description of Related Art

[0003] A security system records a security log in text format when security equipment is used. At this time, in order to defend against security threats it is important to accurately detect an attack script by analyzing the security logs. In this regard, recently, security logs are analyzed using a machine learning-based prediction model to predict an intrusion from the outside. When security logs are analyzed using a machine learning-based prediction model, however, a process of preprocessing the security logs is required.

[0004] Conventional preprocessing models to preprocess security logs may include Word2Vec, Hashing Vectorization, and the like. However, since the conventional technologies analyze relative positions of words in context included in a security log, information included in the security log may be lost or distorted when the security log is preprocessed using the conventional preprocessing model.

[0005] For this reason, there is a demand for a method which can preprocess a security log such that information included in the security log is not lost and distorted and a machine learning-based prediction model can be optimized for analysis of the security log.

SUMMARY

[0006] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0007] The disclosed embodiments are intended to provide an apparatus and method for preprocessing a security log.

[0008] In one general aspect, there is provided an apparatus for preprocessing a security log including a field divider configured to divide a character string of a security log into a plurality of fields on the basis of a structure of the security log; an ASCII code converter configured to convert a character string included in each of the plurality of divided fields into ASCII codes; and a vector data generator configured to generate vector data for each of the plurality of divided fields using the converted ASCII codes.

[0009] The ASCII code converter may convert a predetermined character among a plurality of characters included in the character string into a weighted ASCII code.

[0010] The vector data may include the converted ASCII codes and a length of the character string included in each of the plurality of divided fields.

[0011] The dimension of the vector data may be determined based on a set maximum length of a character string for each of the plurality of divided fields.

[0012] When a specific element among a plurality of elements included in the vector data corresponds neither to the converted ASCII codes nor to the length of the character string included in each of the plurality of divided fields, the vector data generator may set a value of the specific element to be zero on the basis of the determined dimension.

[0013] In another general aspect, there is provided a method of preprocessing a security log including dividing a character string of a security log into a plurality of fields on the basis of a structure of the security log; converting a character string included in each of the plurality of divided fields into ASCII codes; and generating vector data for each of the plurality of divided fields using the converted ASCII codes.

[0014] The converting of the character string may include converting a predetermined character among a plurality of characters included in the character string into a weighted ASCII code.

[0015] The vector data may include the converted ASCII codes and a length of the character string included in each of the plurality of divided fields.

[0016] The dimension of the vector data may be determined based on a set maximum length of a character string for each of the plurality of divided fields.

[0017] The generating of the vector data may include, when a specific element among a plurality of elements included in the vector data corresponds neither to the converted ASCII codes nor to the length of the character string included in each of the plurality of divided fields, setting a value of the specific element to be zero on the basis of the determined dimension.

[0018] Other features and aspects will be apparent from the following detailed description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] FIG. **1** is a diagram illustrating a configuration of a security log analysis system according to one embodiment.

[0020] FIG. **2** is a diagram illustrating an example in which a structure is expressed in a programming language.

[0021] FIG. **3** is a diagram illustrating a configuration of an apparatus for preprocessing a security log according to one embodiment.

[0022] FIG. **4** is a diagram for describing an example in which a character string of a security log is divided into a plurality of fields according to one embodiment.

[0023] FIG. **5** is a diagram for describing an example in which a character string is converted into ASCII codes according to one embodiment.

[0024] FIG. **6** is a flowchart illustrating a method of preprocessing a security log according to one embodiment.

[0025] FIG. **7** is a block diagram for describing a computing environment including a computing device suitable to be used in exemplary embodiments.

[0026] Throughout the drawings and the detailed description, unless otherwise described, the same drawing reference numerals will be understood to refer to the same elements,

features, and structures. The relative size and depiction of these elements may be exaggerated for clarity, illustration, and convenience.

DETAILED DESCRIPTION

[0027] The following description is provided to assist the reader in gaining a comprehensive understanding of the methods, apparatuses, and/or systems described herein. Accordingly, various changes, modifications, and equivalents of the methods, apparatuses, and/or systems described herein will be suggested to those of ordinary skill in the art.

[0028] Descriptions of well-known functions and constructions may be omitted for increased clarity and conciseness. Also, terms described in below are selected by considering functions in the embodiment and meanings may vary depending on, for example, a user or operator's intentions or customs. Therefore, definitions of the terms should be made on the basis of the overall context. The terminology used in the detailed description is provided only to describe embodiments of the present disclosure and not for purposes of limitation. Unless the context clearly indicates otherwise, the singular forms include the plural forms. It should be understood that the terms "comprises" or "includes" specify some features, numbers, steps, operations, elements, and/or combinations thereof when used herein, but do not preclude the presence or possibility of one or more other features, numbers, steps, operations, elements, and/or combinations thereof in addition to the description.

[0029] FIG. **1** is a diagram illustrating a configuration of a security log analysis system **100** according to one embodiment.

[0030] The security log analysis system **100** may be configured with one or more servers for analyzing a security log **110** of the security system to detect an intrusion from the outside and to defend against the intrusion. In this case, the security log **110** may mean log in text form recorded at the time of using security equipment. For example, the security log **110** may include an intrusion detection system (IDS) event log, a web log, a firewall log, an advanced persistent threat (APT) log, and the like. Also, the security log **110** may be in the form of, for example, uniform resource locator (URL).

[0031] Referring to FIG. **1**, the security log analysis system **100** may include a data collection server **101**, a preprocessing server **103**, a learning server **105**, a validation server **107**, and a prediction server **109**.

[0032] The data collection server **101** may collect a security log **110** to be analyzed. Also, the data collection server **101** may design a structure for storing a preprocessed security log. In this case, the structure may include information on a security log preprocessed according to a method described below, for example, a character string included in each of the plurality of fields, a length of the character string included in each of the plurality of fields, vector data for each of the plurality of fields, and the like.

[0033] FIG. **2** is a diagram illustrating an example in which a structure **200** is expressed in a programming language.

[0034] Referring to FIG. **2**, when it is assumed that a character string of a security log is divided into a protocol field **210**, a domain field **220**, a path field **230**, and a query string field **240**, the structure **200** may include information on each of the fields **210**, **220**, **230**, and **240**. For example, the structure **200** may include a character string included in the protocol field **210**, the length of the character string included in the protocol field **210**, and vector data for the protocol field **210** as the information on the protocol field **210**.

[0035] In the above-described example, the structure is designed to include the character string included in the field, the length of the character string included in the field, and the vector data for the field, but is not necessarily limited thereto and may be designed in various forms according to an embodiment.

[0036] Referring back to FIG. **1**, the preprocessing server **103** may preprocess the collected security log **110** into a form suitable to be used as learning data in a prediction model. In this case, the prediction model may be a machine learning-based model that detects an attack script from a specific security log on the basis of learning data including a preprocessed security log. For example, the prediction model may include random forest, K-means clustering, and the like.

[0037] The learning server **105** may train the prediction model based on a predetermined algorithm using the preprocessed security log as learning data. Also, the learning server **105** may re-train the prediction model on the basis of a verification result of the prediction model received from the validation server **107**.

[0038] The validation server **107** may verify the trained prediction model using a validation set. In this case, the validation server **107** may evaluate the performance of the prediction model through the verification of the prediction model.

[0039] The prediction server **109** may detect an attack script from a specific security log using the prediction model.

[0040] FIG. **3** is a diagram illustrating a configuration of an apparatus for preprocessing a security log according to one embodiment.

[0041] The apparatus **300** shown in FIG. **3** for preprocessing a security log may be implemented as, for example, one element of the preprocessing server **103** shown in FIG. **1**.

[0042] Referring to FIG. **3**, the apparatus **300** for preprocessing a security log may include a field divider **310**, an ASCII code converter **330**, and a vector data generator **350**.

[0043] The field divider **310** divides a character string of a security log into a plurality of fields on the basis of a structure of the security log.

[0044] The structure of the security log may be information that a user may obtain by analyzing in advance the general structure of the security log. In this case, the structure of the security log may consist of a plurality of fields. The fields are classified according to the meaning of characters included in the security log. For example, the fields may include a protocol field, a domain field, a path field, a query string field, and the like.

[0045] FIG. **4** is a diagram for describing an example in which a character string of a security log is divided into a plurality of fields according to one embodiment.

[0046] Referring to FIG. **4**, the field divider **310** may divide a character string of a security log **410** into a protocol field **420**, a domain field **430**, a path field **440**, and a query string field **450** on the basis of a structure of the security log **410** that consists of protocol, domain, path, and query string. In addition, the field divider **310** may further divide the divided field into sub-fields on the basis of a structure in which a character string included in the divided field is

repeated. For example, among the plurality of divided fields, the path field **440** may be further divided into split**1** field and split**2** field, and the query string field **450** may be further divided into key**1** field, value**1** field, key**2** field, and value **2** field.

[0047] Referring back to FIG. **3**, the ASCII code converter **330** may convert the character string included in each of the plurality of divided fields into ASCII codes.

[0048] FIG. **5** is a diagram for describing an example in which a character string is converted into ASCII codes according to one embodiment.

[0049] Referring to FIG. **5**, assuming that a character string **520** of "http://" is included in a protocol field **510**, the ASCII code converter **330** may convert each character in the character string **520** included in the protocol field **510** into ASCII codes **530**.

[0050] In addition, assuming that a character string **550** of "samsung.com" is included in a domain field **540**, the ASCII code converter **330** may convert each character in the character string **550** included in the domain field **540** into ASCII codes **560**.

[0051] Referring back to FIG. **3**, according to one embodiment, the ASCII code converter **330** may convert a predetermined character among the plurality of characters included in the character string into a weighted ASCII code.

[0052] In this case, the predetermined character may mean a character generally used in an attach script included in a security log. For example, the predetermined character may be a special character, such as "@," "#," "$," "%," "^".

[0053] The vector data generator **350** generates vector data for each of the plurality of fields using the converted ASCII codes. In this case, the vector data may be data that corresponds to the security log. In addition, the vector data may be learning data to be input to a prediction model.

[0054] According to one embodiment, the vector data may include the converted ASCII codes and a length of the character string included in each of the plurality of divided fields. In this case, the converted ASCII codes and the length of the character string included in each of the plurality of divided fields may be elements of the vector data.

[0055] Referring back to FIG. **5**, a length **580** of the character string **520** included in the protocol field **510** is 7, and thus the vector data generator **350** may generate vector data **570** that includes the ASCII codes **530** converted from the character string **520** included in the protocol field **510** and the length **580** of the character string **520**. At this time, the vector data generator **350** may generate the vector data **570** by placing the ASCII codes **530** at the foremost position and placing the length **580** of the character string **520** at the rearmost position.

[0056] Meanwhile, in the example described above, the vector data generator **350** places the converted ASCII codes at the foremost position and places the length of the character string at the rearmost position to generate the vector data, but is not limited thereto, such that the position of the converted ASCII codes and the position of the length of the character string may be set variously in the vector data.

[0057] Referring back to FIG. **3**, according to one embodiment, the dimension of the vector data may be determined based on a set maximum length of the character string of each of the plurality of divided fields.

[0058] Specifically, a value obtained by adding 1 to the set maximum length of the character string for each of the plurality of fields may be determined to be the dimension of the vector data.

[0059] Referring to FIG. **5**, assuming that the set length of the character string for the protocol field **510** is 8, the dimension of the vector data **570** for the protocol field **510** may be determined to be the nine dimensions.

[0060] Meanwhile, in the example described above, the value obtained by adding 1 to the set maximum length of the character string for each of the plurality of fields is determined to be the dimension of the vector data, but is not limited thereto, and the dimension of the vector data may be determined to be a variety of dimensions.

[0061] Referring back to FIG. **3**, according to one embodiment, when a specific element among a plurality of elements included in the vector data corresponds neither to the converted ASCII codes nor to the length of the character string included in each of the plurality of fields, the vector data generator **350** may set a value of the specific element to be zero on the basis of the determined dimension of the vector data.

[0062] Referring back to FIG. **5**, assuming that the set maximum length of the character string for the protocol field **510** is 8, the dimension of the vector data **570** for the protocol field **510** is determined to be 9 dimensions, and thus there may be a specific element **590** that corresponds neither to the converted ASCII codes **530** nor to the length **580** of the character string, among the plurality of elements included in the vector data **570**. In this case, the vector data generator **350** may generate the vector data **570** by setting a value of the specific element **590** to be zero.

[0063] FIG. **6** is a flowchart illustrating a method of preprocessing a security log according to one embodiment.

[0064] The method illustrated in FIG. **6** may be performed by the apparatus **300** shown in FIG. **3** for preprocessing a security log.

[0065] Referring to FIG. **6**, the apparatus **300** for preprocessing a security log divides a character string of a security log into a plurality of fields on the basis of a structure of the security log (**610**).

[0066] Then, the apparatus **300** for preprocessing a security log converts a character string included in each of the plurality of divided fields into ASCII codes (**620**).

[0067] In this case, the apparatus **300** for preprocessing a security log may convert a predetermined character among a plurality of characters included in the character string into a weighted ASCII code.

[0068] Then, the apparatus **300** for preprocessing a security log generates vector data for each of the plurality of fields using the converted ASCII codes (**630**).

[0069] In this case, when a specific element among a plurality of elements included in the vector data corresponds neither to the converted ASCII codes nor to the length of the character string included in each of the plurality of fields, the apparatus **300** for preprocessing a security log may set a value of the specific element to be zero on the basis of the determined dimension.

[0070] Meanwhile, in the flowchart illustrated in FIG. **6**, the method is described as being divided into a plurality of operations. However, it should be noted that at least some of the operations may be performed in different order or may be combined into fewer operations or further divided into more operations. In addition, some of the operations may be

omitted, or one or more extra operations, which are not illustrated, may be added to the flowchart and be performed.

[0071] FIG. **7** is a block diagram for describing a computing environment including a computing device suitable to be used in exemplary embodiments. In the illustrated embodiments, each of the components may have functions and capabilities different from those described hereinafter and additional components may be included in addition to the components described herein.

[0072] The illustrated computing environment **10** includes a computing device **12**. In one embodiment, the computing device **12** may be one or more components included in the apparatus **300** for preprocessing a security log, such as the field divider **310**, the ASCII code converter **330**, and the vector data generator **350** that are shown in FIG. **3**.

[0073] The computing device **12** may include at least one processor **14**, a computer-readable storage medium **16**, and a communication bus **18**. The processor **14** may cause the computing device **12** to operate according to the above-described exemplary embodiment. For example, the processor **14** may execute one or more programs stored in the computer-readable storage medium **16**. The one or more programs may include one or more computer executable commands, and the computer executable commands may be configured to, when executed by the processor **14**, cause the computing device **12** to perform operations according to an exemplary embodiment.

[0074] The computer-readable storage medium **16** is configured to store computer executable commands and program codes, program data and/or information in other suitable forms. The program **20** stored in the computer-readable storage medium **16** may include a set of commands executable by the processor **14**. In one embodiment, the computer-readable storage medium **16** may be a memory (volatile memory, such as random access memory (RAM), non-volatile memory, or a combination thereof), one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, storage media in other forms capable of being accessed by the computing device **12** and storing desired information, or a combination thereof.

[0075] The communication bus **18** connects various other components of the computing device **12** including the processor **14** and the computer-readable storage medium **16**.

[0076] The computing device **12** may include one or more input/output interfaces **22** for one or more input/output devices **24** and one or more network communication interfaces **26**. The input/output interface **22** and the network communication interface **26** are connected to the communication bus **18**. The input/output device **24** may be connected to other components of the computing device **12** through the input/output interface **22**. The illustrative input/output device **24** may be a pointing device (a mouse, a track pad, or the like), a keyboard, a touch input device (a touch pad, a touch screen, or the like), an input device, such as a voice or sound input device, various types of sensor devices, and/or a photographing device, and/or an output device, such as a display device, a printer, a speaker, and/or a network card. The illustrative input/output device **24**, which is one component constituting the computing device **12**, may be included inside the computing device **12** or may be configured as a device separate from the computing device **12** and be connected to the computing device **12**.

[0077] According to the disclosed embodiments, by preprocessing a security log using ASCII codes, it is possible to convert the security log into vector data without losing or distorting information included in the security log.

[0078] In addition, according to the disclosed embodiment, by converting a security log into vector data, it is possible to improve the performance of a machine learning-based prediction model that analyzes the security log.

[0079] A number of examples have been described above. Nevertheless, it will be understood that various modifications may be made. For example, suitable results may be achieved if the described techniques are performed in a different order and/or if components in a described system, architecture, device, or circuit are combined in a different manner and/or replaced or supplemented by other components or their equivalents. Accordingly, other implementations are within the scope of the following claims.

What is claimed is:

1. An apparatus for preprocessing a security log, comprising at least one hardware processor configured to implement:

a field divider configured to divide a character string of a security log into a plurality of fields on the basis of a structure of the security log;

an ASCII code converter configured to convert a character string included in each of the plurality of divided fields into ASCII codes; and

a vector data generator configured to generate vector data for each of the plurality of divided fields using the converted ASCII codes.

2. The apparatus of claim **1**, wherein the ASCII code converter is further configured to convert a predetermined character among a plurality of characters included in the character string into a weighted ASCII code.

3. The apparatus of claim **1**, wherein the vector data comprises the converted ASCII codes and a length of the character string included in each of the plurality of divided fields.

4. The apparatus of claim **1**, wherein dimension of the vector data is determined based on a set maximum length of a character string for each of the plurality of divided fields.

5. The apparatus of claim **4**, wherein when a specific element among a plurality of elements included in the vector data corresponds neither to the converted ASCII codes nor to the length of the character string included in each of the plurality of divided fields, the vector data generator is further configured to set a value of the specific element to be zero on the basis of the determined dimension.

6. A method of preprocessing a security log, comprising:

dividing a character string of a security log into a plurality of fields on the basis of a structure of the security log;

converting a character string included in each of the plurality of divided fields into ASCII codes; and

generating vector data for each of the plurality of divided fields using the converted ASCII codes.

7. The method of claim **6**, wherein the converting of the character string comprises converting a predetermined character among a plurality of characters included in the character string into a weighted ASCII code.

8. The method of claim **6**, wherein the vector data comprises the converted ASCII codes and a length of the character string included in each of the plurality of divided fields.

9. The method of claim **6**, wherein dimension of the vector data is determined based on a set maximum length of a character string for each of the plurality of divided fields.

10. The method of claim **9**, wherein the generating of the vector data comprises, when a specific element among a plurality of elements included in the vector data corresponds neither to the converted ASCII codes nor to the length of the character string included in each of the plurality of divided fields, setting a value of the specific element to be zero on the basis of the determined dimension.

* * * * *