(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 April 2025 (10.04.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/076509 A1

(51) International Patent Classification:
H04L 9/40 (2022.01) *G06N 3/045* (2023.01)
G06F 21/55 (2013.01) *G06N 3/0455* (2023.01)
G06F 21/56 (2013.01)

(21) International Application Number:
PCT/US2024/050192

(22) International Filing Date:
07 October 2024 (07.10.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/542,708 05 October 2023 (05.10.2023) US

(71) Applicant: **DARKTRACE HOLDINGS LIMITED** [GB/GB]; Maurice Wilkes Building, St John's Innovation Park, Cambridge CB4 0DS (GB).

(72) Inventors: **BAZALGETTE, Timothy**; 12 New Road, Woolmer Green, Knebworth SG3 6JX (GB). **LAL, Jake**; 10 Keynes Road, Cambridge CB5 8PR (GB). **HUMPHREY, Dickon**; 2 Lents Way, Cambridge CB4 1UA (GB). **SELLARS, Phillip**; 62 Windslow House, Greenlane, Trumpington CB2 9DG (GB). **MARTIN, Andrés Curto**; 8 Roger Road, Swaffham Prior, Cambridge CB25 0HX (GB).

(74) Agent: **FERRILL, Thomas S.**; Rutan & Tucker, LLP, 18575 Jamboree Road, 9th Floor, Irvine, CA 92612 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM,

(54) Title: CYBER SECURITY TO DETECT A MALICIOUS FILE

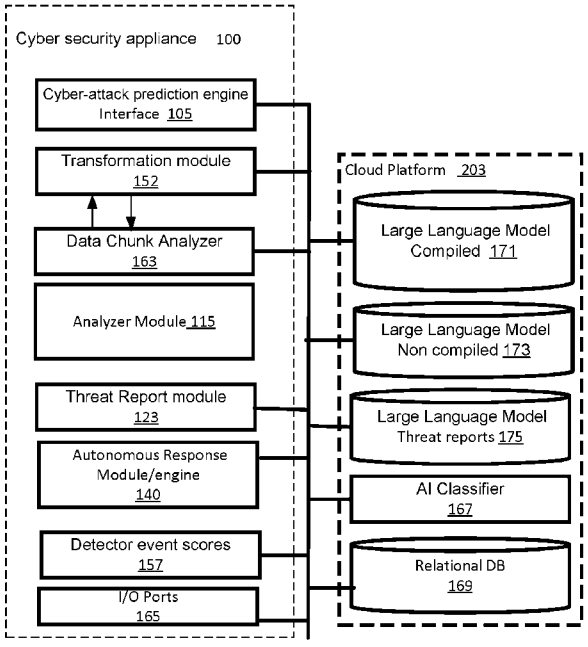

Fig. 1

(57) Abstract: An analyzer module determines whether a file under analysis is likely malicious or not malicious. A transformation module analyzes the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and iii) then to feed the representation of the file under analysis into the LLM. The LLM is trained with MLM to create a semantic understanding of the file that creates a depiction of the file that retains multiple aspects of the information in and behavioral properties about the file as an embedding, in a space that allows the analyzer module to determine whether the file is likely malicious or not malicious via how closely the file under analysis as an embedding is related to a known malicious file or a known not malicious file with similar information and behavioral properties.

WO 2025/076509 A1

[Continued on next page]

DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

CYBER SECURITY TO DETECT A MALICIOUS FILE

NOTICE OF COPYRIGHT

RELATED APPLICATION

[002] This application claims priority under 35 USC 119 to U.S. provisional patent application No. 63/542708, titled "CLOUD-BASED CYBER SECURITY AND METHODS OF OPERATION" filed October 5, 2023, which the disclosure of such is incorporated herein by reference in its entirety.

FIELD

[003] Cyber security and in an embodiment use of Artificial Intelligence in cyber security.

BACKGROUND

[004] Cybersecurity attacks have become a pervasive problem for enterprises as many computing devices and other resources have been subjected to attack and compromised. A "cyberattack" constitutes a threat to security of an enterprise (e.g., enterprise network, one or more computing devices connected to the enterprise network, or the like). A cyber threat from a cyberattack may involve malicious software, an insider attack, and other threat introduced into a computing device and/or the network. The cyber threats may further represent malicious or criminal activity, ranging from theft of credential to even a nation-state attack, where the source initiating or causing the security threat is commonly referred to as a "malicious" source.

[005] Malicious files are traditionally analyzed using hashes. Though these are ideally sorted to determining if a specific file has been seen before, hashing algorithms have a number of properties that make them less than ideally suited for this analysis. In particular, they are lossy, removing almost all information associated with the file, and

they are explicitly designed for two similar files to have entirely distinct hashes. Consequently, attackers often take advantage of these deficiencies, changing a few bytes of a malicious payload to give it a new hash and make it undetectable by traditional antivirus software.

[006] Various approaches have attempted to address these weaknesses of hash algorithms for this purpose. For instance, there exist fuzzy hashing algorithms such as SSDEEP, which perform piecewise hashing to attempt to give similar files similar hashes. While this improves one problematic feature of hashing algorithms, the approach is nevertheless insufficiently aware of file structure, and so has mixed results. Other approaches include representation of files as images, and using convolutional neural networks (typically used for image analysis) to assess them. While this is more performant, the focus is often specific to the structure of the executable, and models are often trained as supervised classifiers, requiring labelled datasets, and limiting their use cases.

SUMMARY

[007] Methods, systems, and apparatus are disclosed for an Artificial Intelligence-based cyber security system. In an embodiment, the cyber security system can include an analyzer module, a transformation module, a large language model (LLM), and other components. The analyzer module determines whether a file under analysis is likely malicious or not malicious. The transformation module analyzes the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and iii) then to feed the representation of the file under analysis into the LLM. The LLM is trained with masked language modelling to create a semantic understanding of the file under analysis that creates a depiction of the file that retains multiple aspects of the information in and behavioral properties about the file under analysis as an embedding, in a space that allows the analyzer module to determine whether the file under analysis is likely malicious or not malicious via how closely the file under analysis as an embedding is related to a known malicious file or a known not

malicious file with similar information and behavioral properties, without having to generate some kind of hash.

[008] These and other features of the design provided herein can be better understood with reference to the drawings, description, and claims, all of which form the disclosure of this patent application.

DRAWINGS

[009] The drawings refer to some embodiments of the design provided herein in which:

[010] Figure 1 illustrates a block diagram of an embodiment of an example transformation module in the cyber security appliance configured to analyze the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and iii) then to feed the representation of the file under analysis into a Large Language Model (LLM) in the cloud platform.

[011] Figure 2 illustrates a block diagram of an embodiment of an example transformation module in the cyber security appliance configured to analyze the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and iii) then to feed the representation of the file under analysis into a Large Language Model (LLM) in the cyber security appliance in order for the analyze module to determine whether the file under analysis is likely malicious or not malicious.

[012] Figure 3 illustrates a graph of an embodiment of an example chain of unusual behavior for, in this example, the email activities and IT network activities deviating from a normal pattern of life in connection with the rest of the system/network under analysis.

[013] Figure 4 illustrates a block diagram of an embodiment of the AI-based cyber security appliance with example components making up a detection engine that protects a system, including but not limited to a network/domain, from cyber threats.

[014] Figure 5 illustrates a diagram of an embodiment of i) the cyber threat detection engine using Artificial Intelligence algorithms trained to perform a first machine-learned task of detecting the cyber threat, ii) an autonomous response engine using Artificial Intelligence algorithms trained to perform a second machine-learned task of taking one or more mitigation actions to mitigate the cyber threat, iii) a cyber-security restoration engine using Artificial Intelligence algorithms trained to perform a third machine-learned task of remediating the system being protected back to a trusted operational state, and iv) a cyber-attack prediction engine using Artificial Intelligence algorithms trained to perform a fourth machine-learned task of Artificial Intelligence-based simulations of cyberattacks to assist in determining 1) how a simulated cyberattack might occur in the system being protected, and 2) how to use the simulated cyberattack information to preempt possible escalations of an ongoing actual cyberattack, in order for these four Artificial Intelligence-based engines to work together.

[015] Figure 6 illustrates a block diagram of an embodiment of the cyber-attack prediction engine with Artificial Intelligence-based simulations conducted in the cyber-attack prediction engine by constructing a graph of nodes of the system being protected (e.g. a network) including i) the physical devices connecting to the network, any virtualized instances of the network, user accounts in the network, email accounts in the network, etc. as well as ii) connections and pathways through the network to create a virtualized instance of the network to be tested.

[016] Figure 7A illustrates a diagram of an embodiment of the cyber-attack prediction engine and its Artificial Intelligence-based simulations constructing an example graph of nodes in an example network and simulating how the cyberattack path might likely progress in the future tailored with an innate understanding of a normal behavior of the nodes in the system being protected and a current operational state of each node in the graph of the protected system during simulations of cyberattacks.

[017] Figure 7B illustrates a diagram of an embodiment of the cyber-attack prediction engine and/or the cyber-attack restoration engine assigning scores for a portion of the graph of nodes of the system being protected (e.g. a network) including i) the physical devices, accounts, etc. in the system, etc. as well as ii) connections and attack pathways through the network.

[018] Figure 8 illustrates a block diagram of an embodiment of the AI-based cyber security appliance with the security awareness training system and other Artificial Intelligence-based engines plugging in to protect a system.

[019] Figure 9 illustrates a block diagram of an embodiment of one or more computing devices that can be a part of the Artificial Intelligence-based cyber security system including the multiple Artificial Intelligence-based engines and the security awareness training system discussed herein.

[020] Figure 10 illustrates an embodiment of an example graph of probability distributions used by the detector for events with initial scores equal to 0%, 25%, 50%, 75% and 100%.

[021] Figure 11 illustrates an embodiment of an example table used by the detector showing an initial score, rarity counts across the fleet of cyber security appliances, and modified score for various examples using synthetic data.

[022] While the design is subject to various modifications, equivalents, and alternative forms, specific embodiments thereof have been shown by way of example in the drawings and will now be described in detail. It should be understood that the design is not limited to the particular embodiments disclosed, but – on the contrary – the intention is to cover all modifications, equivalents, and alternative forms using the specific embodiments.

DESCRIPTION

[023] In the following description, numerous specific details are set forth, such as examples of specific data signals, named components, number of servers in a system, etc., in order to provide a thorough understanding of the present design. It will be apparent, however, to one of ordinary skill in the art that the present design can be practiced without these specific details. In other instances, well known components or methods have not been described in detail but rather in a block diagram in order to avoid unnecessarily obscuring the present design. Further, specific numeric references such as a first server, can be made. However, the specific numeric reference should not be interpreted as a literal sequential order but rather interpreted that the first server is different than a second server. Thus, the specific details set forth are merely exemplary. Also, the features implemented in one embodiment may be implemented in

another embodiment where logically possible. The specific details can be varied from and still be contemplated to be within the spirit and scope of the present design.

[024] Figure 1 illustrates a block diagram of an embodiment of an example transformation module in the cyber security appliance configured to analyze the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and iii) then to feed the representation of the file under analysis into a Large Language Model (LLM) in the cloud platform. Figure 2 illustrates a block diagram of an embodiment of an example transformation module in the cyber security appliance configured to analyze the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and iii) then to feed the representation of the file under analysis into the LLM in the cyber security appliance in order for the analyze module to determine whether the file under analysis is likely malicious or not malicious.

[025] The transformation module 152 analyzes the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and ii) then to put the representation of the file into a consistent file format respective to other previous files that have been analyzed, and iii) then to feed the representation of the file under analysis into a LLM 171, 173. Again, the transformation module 152 puts all files fed into the LLM 171, 173 into a similar file manner representation that the LLM 171, 173 trained by masked latent semantic modelling takes in (including its content, behavior, etc.) in the representation of the file under analysis.

[026] The transformation module 152 sends the representation of the file under analysis across a secure connection on the Internet to the LLM 171, 173.

[027] The data chunk analyzer 163 cooperates with the LLM 171, 173 and the transformation module 152. The data chunk analyzer 163 can be 1) trained with machine learning or 2) scripted in software code to identify different parts of the file under analysis, and then break the file under analysis (and/or the representation of that file) into its different parts when a size of the representation of that file under analysis

will not fit into a content window size requirement of the LLM 171, 173. The data chunk analyzer 163 can i) determine a size requirement of a content window of the LLM 171, 173 and ii) put portions of the representation of that file under analysis into chunks/portions sized small enough to fit into the size requirement of the content window of the LLM 171, 173 in a sequence such that the LLM 171, 173 can retain a context that these portions sized small enough to fit into that content window size requirements all belong to a same file. The data chunk analyzer 163 can communicate with an API or other manner of the LLM 171, 173 to determine its content window size limit and/or merely be programmed with the LLM 171, 173's content window size limit. Thus, the data chunk analyzer 163 can take portions of the file under analysis and put them into the content window size of the input of the LLM 171, 173. The data chunk analyzer 163 is scripted to examine a file and/or representation of that file and identify the different parts of a file and/or representation of that file and break it into different parts of that file. The transformation module 152 and the data chunk analyzer 163 cooperate to understand a file under analysis and then break down the file under analysis into particular parts. Each part can be broken down into its particular part when the size is greater than the amount of bytes allowed by the content window size of the LLM 171, 173 so that it can be fed into the content window. The broken down parts are successively identified by the data chunk analyzer 163 and fed into the LLM 171, 173 content window successively allowing the total size of the file and/or representation of that file to exceed the size of the content window size of the LLM 171, 173. The transformation module 152 as well as the data chunk analyzer 163 can be trained with machine learning or scripted with a set of algorithms to identify certain parts of a file and natural portions to break down portions of a file under analysis. The data chunk analyzer 163 and the transformation module 152 are scripted to examine the file and/or representation of that file to work out where natural breaks are in, for example, the wording, section titles, page breaks, etc. so that the LLM 171, 173 will be able to still retain the context being conveyed by the file when it is fed to the Large Language Model 171, 173. For example, as long as the data chunk analyzer 163 feeds the content window of the LLM 171, 173 several paragraphs at a time, then the LLM 171, 173 will not lose the context of the file and/or representation of that file. However, if the data

chunk analyzer 163 feeds too much content, such as simply based upon byte size to match the same max size in the content window, then you might lose some important representation in that file. Simply feeding a higher level representation of the file under analysis at merely the halfway size of that file without accounting for how that file is naturally structured could cause you to lose some important representation in that file. In an embodiment, the data chunk analyzer 163 is a part of the transformation module 152. In an embodiment, when the file is small enough to fit inside the size of the content window size of the LLM 171, 173, then the transformation into the representation fed into the LLM 171, 173 is a single operation rather breaking down the file.

[028] Also, a raw file by itself (versus a representation that includes a simplified summary on information in and behavioral properties about the file) can be too much data for the LLM 171, 173 to evaluate all at once. For example, an example file could be 2 MB of data, which is a lot of information. 2 MB of data could be greater than the content window size limit of the LLM 171, 173. Moreover, 2 MB of raw data without direction of its structure could be a lot of information for the LLM 171, 173 to digest. Accordingly, the LLM 171, 173 could take a very long time to learn what parts are relevant and which bits are not relevant in 2 MBs of raw data without the direction of its structure. Instead, the LLM 171, 173 cooperates with the transformation module 152 so that all of the files analyzed are preprocessed in a similar way to automate recognizing and subsequently identifying the file in its simplified representation to the LLM 171, 173.

[029] The simplified summary on information in and behavioral properties about the file under analysis can include the type of file under analysis, break down files under analysis in a same repetitive algorithmic way, and feed those files into the LLM 171, 173 in the same way and in a same file format. Further, when the LLM 171, 173 and the transformation module 152 couple with the data chunk analyzer 163, then the size of data chunks is fed in a way that the LLM 171, 173 allows the LLM 171, 173 to efficiently and consistently perform its analysis to gain a semantic understanding of the file under analysis. The LLM 171, 173 does not just analyze a raw file, with all of the different types of files, potential contents in a file, potential different software languages that could be utilized to create/script that file, etc.

[030] For example, transformation module 152 may determine that the file is an executable file. The type of file could be an executable file, which then will have a behavioral profile extracted on how that file might make all of the system calls that file makes and some other information. The transformation module 152 can be scripted to perform the actions and steps that mimic something that experts who reverse engineer software code often do and work out for executable files. Thus, the scripts can be based upon following the steps that software engineers who are experts in reverse engineering and understanding files take in order to try and understand which bits are relevant, and then ideally get to some representation that shows all those relevant things. However, the script also captures/extracts behavioral data of how that file operates.

[031] In an example, transformation module 152 may determine that the file is coded in Python or compiled C+ and the LLM 171, 173 trained with masked latent semantic modelling can train on different kinds of representations. In an example, a file may have a difference in size, such as a small sized Python script that is not compiled and contains a small write up to help explain exactly what the file is doing; whereas, an example compiled C+ executable or other compiled file will be a large number of bytes, and that will be encoding different assembly codes for the system that the files are running on. Note, a secondary check is made to check the software code itself and see if that software code matches up to the provided description of the code. A well-known attack vector for causing LLMs to misclassify software scripts is based on the malicious actor intentionally providing an incorrect description of the software code to fool LLMs. Thus, checking the software code itself provides an accurate analysis of the nature of the file under analysis and can add another factor to determine whether that file is malicious when the analysis of the nature of software code itself does not match the provided description of the code. A compiled file in C+ is going to be much bigger, with more information and generally much harder to spot the crucial bits of information that are within the file.

[032] The LLM 171, 173 cooperating with the transformation module 152 and the AI classifier 167 creates another channel of threat intelligence when the cyber security system sees a new malicious file, identified by any cyber security appliance 100

in the fleet in any domain e.g. an IT network domain, an email domain, a cloud domain, etc. The LLM 171, 173 cooperating with the transformation module 152 and the AI classifier 167 can use the new malicious file to create new embedding that an analyzer module 115 in a local cyber security appliance 100 can then use and detect any similar files from affecting any of the networks protected by this local cyber security appliance 100 or other the cyber security appliances in the fleet. The transformation module 152 can upload and send the representation of the file under analysis over a secure channel in a network, such as the Internet, to a cloud platform 203 when the characteristics of the file under analysis are not similar to known not malicious files with similar information and behavioral properties. Thus, new files that are not similar to known good files are sent for further analysis by the LLM 171, 173 and AI classifier 167. The cloud platform 203 is configured to host the LLM 171, 173 and an AI classifier 167 and/or the relational database 169 to create the embedding and categorize the embedding as i) not a malicious file or ii) a malicious file when a new file is identified. The cloud platform 203 is configured to send the embedding and categorization of the embedding down across a network, such as the Internet, to a plurality of local networks, each local network with its own cyber security appliance 100 protecting that local network so that the analyzer module 115 in its own cyber security appliance 100 can determine whether the file under analysis is likely malicious or not malicious. The analyzer module 115 can use a local instance of the AI classifier 167 to evaluate new files under analysis to trained upon embeddings of known good and known malicious files. The cloud platform 203 regularly trains and updates the AI classifier 167 in the cloud platform 203 and then updates the local instances of the AI classifier 167. Each local cyber security appliance 100 can store its own embeddings of files for comparison. In an embodiment, the local cyber security appliance 100 could have all of the new files (and thus no filtering out of non-suspicious routine files) under analysis sent back to a cloud platform 203, where the LLM 171, 173 operates and the embeddings are created. In an embodiment, the local cyber security appliance 100 can contain the LLM 171, 173, the transformation module 152, and the AI classifier 167 but the size of the LLM 171, 173 needs to be limited in size and power consumption. The transformation module 152 can create a simplified summary of the new file under analysis for a reduced

bandwidth to send all of those simplified summary files over a network. Although in some domains such as email, the sensor hook in the email pipeline can feed a copy of the exact file. However, in SaaS and/or file transfer, a representation of the file under analysis - being the link to the actual file, is sent so that the actual copies of the file can be downloaded and analyzed by an LLM 171, 173 with Masked Language Modeling (MLM) trained on analyzing email files.

[033] Masked Language Modeling Training

[034] The LLM 171, 173 trained by MLM can have an initial training similar to building a foundation for the LLM 171, 173 train on lots and lots of representations of different files; rather than documents from the Internet, are used as the training data. The MLM may be masked latent semantic modelling. The LLM 171, 173 trained by MLM trains on representations of files. The LLM 171, 173 trained by MLM uses the masked language modeling technique to build the model, build an understanding of what things mean, and what the files look like in general.

[035] Note, that the MLM is a type of self-supervised learning, which means the model learns from the input files and any data supplied about the file itself without the need for explicit labels or annotations added by some human onto that file.

[036] The transformation module 152 and the data chunk analyzer 163 cooperate to reverse engineer a large amount of files under analysis to create a big corpus of high level representations of files, which creates a training data set, which the system will use to train the LLM 171, 173 from scratch as if it was a foundational model. The transformation module 152 performs a data transformation step into training data which can then be used to train this LLM 171, 173 from scratch.

[037] The system can use MLM or some machine learning technique to train the LLM 171, 173 on each representation of its corresponding file, and then that trained model means that the LLM 171, 173 produces an embedding of each file, in the same way that a Large Language Model 171, 173 trained on documents can give you embeddings of documents and have this embedding representation of the file's semantic contents. An embedding can be a continuous vector representations of words, phrases, entire texts, and/or tokens about the file that capture their semantic meanings in a high-dimensional space to provide contextual awareness of the meaning

of the words, phrases, entire texts, behaviors, and/or tokens in the embedding. The LLM 171, 173 is trained with MLM to create a semantic understanding of the file under analysis that creates a meaningful depiction of the file that retains multiple aspects of the information in and behavioral properties about the file under analysis as an embedding, in a space that allows the analyzer module 115 to determine whether the file under analysis is likely malicious or not malicious via how closely the file under analysis as an embedding is related to a known malicious file or a known not malicious file with similar information and behavioral properties, without having to generate some kind of hash. The embedding of a file explains what behavioral properties that file is doing, what its properties are, what its contents/information are within the file, etc.; and thus, a semantic understanding of that file. This semantic understanding can be used as a superior alternative to hashes, which allows for intelligent, fuzzy matching, and as a foundation model for analysis of files for other purposes, such as classification, and is robust against minor modifications and by malicious actors in a way that existing techniques are not yet. The semantic understanding is more robust than a hash because it retains a lot of information about the file and does not lose as much information as a hash. Semantics can be a study of a particular component's meaning and how that meaning is created. The semantic representation is a direct result of the embedding produced by the MLM training process.

[038] In the MLM a couple of processes occur such as masking occurs, one or more predictions occur, and machine learning occurs. In masking, a portion (e.g. aspects) of the input file can be replaced with a token, such as a mask. In MLM, the system can drop random tokens and ask the LLM 171, 173 to guess what those tokens are. In prediction, the LLM 171, 173 then uses the context of the other parameters and meta data about the file under analysis to make one or more predictions, such as predicting both a category of file and other aspects of the file. The semantics can be defined using a mathematical theory. In learning, the LLM 171, 173 learns the relationships between the files under analysis, about various aspects including information in and behavioral properties about the file under analysis, and potentially how they fit into the context of malicious or not malicious within that category of files under analysis in order to identify malicious files. The LLM 171, 173 with MLM provides

a robust way of identifying malicious files over a traditional hash based system, based upon previous information that does not require seeing exactly the same files again. MLM of files under analysis provides a way of obtaining a proper semantic understanding of what a file is, and what properties that file has.

[039] In the training pipeline, a corpus of known bad/ malicious files and a corpus of known good files, which are reduced to representations by the transformation module 152, which are then fed in the LLM 171, 173 undergoing training with mask language modelling. The transformation module 152 is scripted to analyze a file and reverse engineer to break that file into its constituent parts, meta data, and behavior and then turn each representation into a similar format to analyze based upon the file type. The transformation module 152 is scripted to apply that same process that transforms the file into a representation to input to the model and to use those representations as training data for the LLM 171, 173 that we train from scratch. Once the LLM 171, 173 is trained, then the LLM 171, 173 can be deployed and have the transformation module 152 feed in new files including potentially bad/malicious executable files that we are seeing in the wild and let the LLM 171, 173 trained with MLM to produce an embedding understanding of that file, which still retains a lot of information specific about that file under analysis.

[040] The LLM 171, 173 is largely trained with unsupervised techniques such as MLM using unlabeled training datasets, allowing the LLM 171, 173 to learn the structure of large volumes of data, resulting in a foundation model which can be used for embeddings, generation, and classification. By representing a file in a textual form that is a representation of that file, a “large file model” could be trained on a large number of files, providing a model which can intelligently represent files as vector embeddings, as well as be adaptable for other tasks such as classification. Moreover, these textual representations of files need not necessarily be raw bytes but could also include the output of more complex static or even active (e.g., sandbox-based) analysis of an executable file.

[041] The training may consist of a number of training and data cycles. For example, a training set of files needs to be a large set of files in the order of millions of

different types of files. Each of those files will have to have a large number of tokens. Each training cycle will consist of a number of cycles such as a few epochs.

[042] In an embodiment, the AI classifier could most likely be a fine-tuned version of the LLM itself with a classification head, rather than a distinct model.

[043] Deployment

[044] The resulting LLM 171, 173 trained with MLM is then able to encode all these insights from analysis into its embedding understanding of that file under analysis. The embeddings are clustered together with similar files, such as similar executable files, based on non-trivial components of their structure, information in, and behavioral properties/ activity about the file under analysis, rather than on exact byte sequences.

[045] The LLM 171, 173 trained with MLM analyzes the content, meta data, category of the file, behavior of the file under analysis, as well as additional points (e.g. if it is a spreadsheet, Word doc, pdf, email, etc. OR a compiled file such as an executable file) to produce a machine learning embedding on the file under analysis, (e.g. malicious file or potentially malicious file). The masked LLM 171, 173 analyzes the content, meta data, and behavior of the file under analysis to extract information, (e.g. meta data on the file, context from the file, category of the file, as well as additional points, for example, if it's an executable file on what other files within the computing system the executable file makes communications with and tries to interact with) in order to produce an embedding understanding of that file under analysis.

[046] In general, the representation of the file with summaries of content, meta data, and behavior of a file retains most of the essential information associated with the file. Thus, when analyzed by the LLM 171, 173 and encapsulated into a produced embedding, then the embedding compared to another embedding or similar understanding of a file is more easily matched when comparing two similar files than with a hash.

[047] In an embodiment, the LLM 171, 173 trained by MLM to take in the representation of the file can be implemented with multiple specifically trained LLMs 171, 173, each on its own specific type of file. For example, i) a first LLM 171 can be trained to analyze compiled files under analysis, ii) a second LLM 173 can be trained to analyze non-compiled files under analysis, iii) a third LLM can be trained to analyze

other specific types of files, etc. The transformation module 152 can be configured to determine whether the file under analysis is compiled or not compiled and then to send the representation of the file under analysis to the first LLM 171 trained to analyze compiled files or the second LLM 173 trained to analyze non-compiled files under analysis, as appropriate.

[048] Note, the transformation module 152 is scripted to determine whether a file under analysis is compiled or not compiled because if the file is compiled then lots of information the system will not necessarily have access to, for example, the file's actual source code. Most executable files will be, for example, an .exe, .run, etc. but is always a compiled file. Each executable file will have a behavioral profile extracted on how that file might make all of the system calls, (e.g. files and other components that the file under analysis makes communications with and tries to interact with), what is the file trying to do, where that file is executing, sequences of how and to what the file will make all of the system calls, and other behavior of that executable file, determine the type of software language that the file is coded in, what computing platform the executable file executes on, and other files within the computing system, etc.

[049] A further step is that the analyzer module 115 can have a local instance of the AI classifier 167 and/or cooperate with the AI classifier 167 implemented on the cloud platform 203. The analyzer module 115 with the AI classifier 167 can use that that embedding, instead of a hash, to identify a nature of files through comparison with known malicious and/or known not malicious file as well as be used to find similar files. The AI classifier 167 can readily understand how similar a semantic understanding in an embedding is to other files.

[050] The LLM 171, 173 can feed the produced embedding on the file under analysis to at least one of 1) an AI classifier 167 that is trained to examine the embedding to determine whether the file under analysis is likely a malicious file or not likely a malicious file or 2) a relational database 169 that is configured to store the embedding in a first logical area/cluster of relatedness from two or more different logical areas of subject matter and relationships within the relational database 169. Next, the AI classifier 167 and/or the relational database 169 are then configured to cooperate with the analyzer module 115 to determine whether the produced embedding is put and

stored in a category bucket designated as likely a malicious file or not likely a malicious file.

[051] Next, a trained AI classifier 167 on analyzing embeddings can put the produced embedding into a category of, for example, a likely malicious file or not likely a malicious file based upon how similar the embedding of the new file under analysis is to the embedding of existing files already in one category or the other category. The nature/behavior of a malicious file will be similar generally even if the creator changes the load or content within that file to avoid detection. Alternatively, even if a nature/behavior of a malicious file is scripted to be changed to avoid detection from earlier versions of the malicious file, then the new malicious file may still make similar calls and have similar signature meta data and content to an earlier version of the malicious file, and then still be closely clustered to a known malicious file or inversely a known safe/normal file.

[052] A BAYESIAN METHOD TO DYNAMICALLY IMPROVE EVENT SCORES WHEN NEW DATA IS AVAILABLE

[053] The detector 157 utilized by the analyzer module 115 can use a Bayesian statistical inference approach to dynamically improve how abnormal event scores contribute to whether the event is an indicator of a cyber threat when new data is available, based upon factoring in data collected across a fleet of cyber security appliances, each cyber security appliance containing its own analyzer module 115.

[054] In an embodiment, the detector 157 deployed within a cyber security system is configured to automatically generate an alert based on the analysis of the external traffic activity produced by a device that is connected to an enterprise network. For example, a device producing suspicious regular connections to the external hostname example[.]com might generate an alert with an initial score equal to 80%.

[055] Operability of the cyber security software can provide extra and relevant information on the interactions between this hostname and other enterprise networks monitored by sensors. If the external endpoint example[.]com is popular across many networks, it is potentially more likely that the connections to this endpoint are part of legitimate processes. Therefore, this extra information can be used to reduce the initial score of the associated alert, and, in some cases, the alert might be even suppressed.

Conversely, if the hostname is very rare across many different enterprise networks, the chances that the connectivity is suspicious are higher and therefore it makes sense to increase the final score, for example from 80% to 95%.

[056] Method

[057] The detector 157 uses a statistical inference approach based on the Bayes theorem to dynamically update a probability hypothesis when new extra information is available. The detector 157 starts with a preliminary hypothesis P(S0) that establishes how likely a given event involving a hostname example[.]com is going to have a score S0 within the [0, 1] interval. In the detector 157, a beta distribution function using the appropriate parameters is selected (see Figure 10). Figure 10 illustrates an embodiment of an example graph of probability distributions used by the detector 157 for events with initial scores equal to 0%, 25%, 50%, 75% and 100%.

[058] Using standard Bayesian inference, the probability P(S0|Ri) that an event involving example[.]com has a score S0 based on new extra data Ri across different enterprise networks can be determined following the next expression:

[059] $P(S!|R'') = P(S!)P(R''|S!)\ P(R'')$

[060] P(S0) is the prior or initial probability that the event has a score S0.

[061] P(Ri|S0) is the likelihood or probability that example[.]com has Ri for a given score S0.

[062] P(Ri) is a normalization factor which only depends on Ri.

[063] The application of this score modification approach will return score modifications similar to the ones presented in Table 1 (See Figure 11), where some clearly legitimate events have their initial scores reduced and other unclear events have their initial scores increased. Figure 11 illustrates an embodiment of an example table (table 1) used by the detector 157 showing an initial score, rarity counts across the fleet of cyber security appliances (e.g. fleet wide analysis), and modified score for various examples using synthetic data.

[064] The detector 157 is also able to self-learn and adapt as the different number of sensors providing data to the cyber security software increases over the time. For example, if example[.]com is seen in 76 sensors and there are a total of 150 sensors sending data to the cyber security software, the endpoint will be considered

popular, and the score will likely be decreased. If the endpoint is seen in 76 sensors and there are a total of 15,000 sensors sending data to the cyber security software, the endpoint will be considered rare, and the score will likely be increased.

[065] EXTRACTION OF BEHAVIORAL DATA/INFORMATION FROM THREAT INTELLIGENCE REPORTS TO MATCH CYBER SECURITY INCIDENTS

[066] A threat report module 123 to use an LLM 175 trained to analyze a threat intel report, generated by third parties, and extract behavioral data in order to 1) match the extracted behavioral data to data currently present in a network under analysis to detect whether a similar potential cyber security incident is present in the network under analysis and 2) to use the extracted behavioral data as a basis for a generation of a security incident simulation performed by a prediction engine. The prediction engine 105 can use Artificial Intelligence algorithms trained to perform a machine-learned task of Artificial Intelligence-based simulations of cyberattacks on the network under analysis.

[067] A user of the system can ask the threat report module 123 when there are signs that they have been compromised by an attacker published in the latest cyber threat report. The threat report module 123 is able to match reported cyber threats to signs seen on the user's network; and thus, cyber security appliance provides an answer to this request. The threat report module 123 is able to generate simulations with the use of the prediction engine 105, which mimic the events described in a threat report so users can practice and prepare for reported threats. An example, threat intelligence report could be, for example - Mandiant, has written up, or something like that, which generally describes the cyberattack, the approaches that a malicious actor has taken through the life cycle of an attack. Threat intelligence reports have a title, a body, and text, and then authors will sometimes write them out in bullet points. The threat report module 123 can use the LLM 175 trained to analyze a threat intel report and extract behavioral data and put them into a format that is understandable by users (e.g. generally by a MITRE attack).

[068] These threat reports can contain descriptions of the tactics, techniques, and procedures (TTPs) that the threat actor has carried out in a particular compromise. Often these descriptions are in unstructured text and therefore are not easily matched

by a computer to other observations. Using the LLM 175 trained to analyze a threat intel report and extract behavioral data, such a model can be used to match free text descriptions to defined threat categories. For example, sentences describing a scanning activity could be matched to the specific threat category in MITRE's ATT&CK framework.

[069] Using this technique, any well described report could be analyzed to extract a series of behaviors in a structured format. Alternatively, some reports may include relevant threat categories, but in an undefined structure which could be identified and extracted.

[070] The extracted threat intel can be converted to a vector representation and searched across a vector database of AI Analyst incidents to find likely matching incidents. Alternatively, a database of behavioral vectors could be built up from numerous cyber threat reports and searched to provide attribution for vectorized AI Analyst incidents.

[071] Extracted TTPs could be used to generate training simulations for known threat behavior. TTP would be extracted from a set of reports relating to a particular threat actor. These would be fed into a graph generation ML model (trained on previous AI Analyst incidents). The graph generation model would then suggest a likely graph representing the inputted TTPs. The suggested graph can be fed into the prediction engine 105 simulation architecture and realistic simulations of known threats could be produced.

[072] The autonomous response engine 140 can cooperate with the analyzer module 115 to perform one or more mitigation actions to mitigate a cyber threat caused by the malicious file without a need for a human to initiate the mitigation actions. The autonomous response engine 140 can be 1) trained with machine learning, 2) scripted in software code, or 3) a combination of both to perform one or more mitigation actions.

[073] **Additional Details**

[074] The following text below discusses how some of the other components in the cyber security system operate; and thus, how these components respond to the commands, requests, and communications from the system.

[075] Figure 4 illustrates a block diagram of an embodiment of the AI-based cyber security appliance with example components making up a detection engine that protects a system, including but not limited to a network/domain, from cyber threats. Various Artificial Intelligence models and modules of the cyber security appliance 100 cooperate to protect a system, such as one or more networks/domains under analysis, from cyber threats. In an embodiment, the AI-based cyber security appliance 100 may include a trigger module, a gather module 110, an analyzer module 115, a cyber threat analyst module 120, an assessment module 125, a user interface and formatting module 130, a data store 135, an autonomous response engine 140 and/or an interface to an autonomous response engine 140, an Information Technology network domain module 145, an email domain module 150, and a coordinator module 155, one or more AI models 160 (hereinafter, AI model(s)"), and/or other modules. The AI model(s) 160 may be trained i) with machine learning on a normal pattern of life for entities in the network(s)/domain(s) under analysis, ii) with machine learning on cyber threat hypotheses to form and investigate a cyber threat, iii) on what are a possible set of cyber threats and their characteristics, symptoms, remediations, etc., an interface to a restoration engine 190, an interface to a cyber-attack prediction engine 105, and other similar components.

[076] The cyber security appliance 100 can host the cyber threat detection engine and other components. The cyber security appliance 100 includes a set of modules cooperating with one or more Artificial Intelligence models configured to perform a machine-learned task of detecting a cyber threat incident. The detection engine uses the set of modules cooperating with the one or more Artificial Intelligence models in the cyber security appliance 100 to prevent a cyber threat from compromising the nodes (e.g. devices, end users, etc.) and/or spreading through the nodes of the network being protected by the cyber security appliance 100.

[077] The cyber security appliance 100 with the Artificial Intelligence (AI)-based cyber security system may protect a network/domain from a cyber threat (insider attack, malicious files, malicious emails, etc.). The cyber security appliance 100 can protect all of the devices on the network(s)/domain(s) being monitored. For example, the IT network domain module (e.g., first domain module 145) may communicate with network

sensors to monitor network traffic going to and from the computing devices on the network as well as receive secure communications from software agents embedded in host computing devices/containers. Other domain modules such as the email domain module 150 and a cloud domain module operate similarly with their domain. The steps below will detail the activities and functions of several of the components in the cyber security appliance 100.

[078] The gather module 110 may be configured with one or more process identifier classifiers. Each process identifier classifier may be configured to identify and track one or more processes and/or devices in the network, under analysis, making communication connections. The data store 135 cooperates with the process identifier classifier to collect and maintain historical data of processes and their connections, which is updated over time as the network is in operation. In an example, the process identifier classifier can identify each process running on a given device along with its endpoint connections, which are stored in the data store 135. In addition, a feature classifier can examine and determine features in the data being analyzed into different categories.

[079] The analyzer module 115 can cooperate with the AI model(s) 160 or other modules in the cyber security appliance 100 to confirm a presence of a cyber threat in cyberattack against one or more domains in an enterprise's system (e.g., see system/enterprise network 791, 792, and 747 of Figure 6). A process identifier in the analyzer module 115 can cooperate with the gather module 110 to collect any additional data and metrics to support a possible cyber threat hypothesis. Similarly, the cyber threat analyst module 120 can cooperate with the internal data sources as well as external data sources to collect data in its investigation. More specifically, the cyber threat analyst module 120 can cooperate with the other modules and the AI model(s) 160 in the cyber security appliance 100 to conduct a long-term investigation and/or a more in-depth investigation of potential and emerging cyber threats directed to one or more domains in an enterprise's system. Herein, the cyber threat analyst module 120 and/or the analyzer module 115 can also monitor for other anomalies, such as model breaches, including, for example, deviations for a normal behavior of an entity, and other techniques discussed herein. The analyzer module 115 and/or the cyber threat analyst module 120 can cooperate with the AI model(s) 160 trained on potential cyber threats in order to assist in

examining and factoring these additional data points that have occurred over a given timeframe to see if a correlation exists between 1) a series of two or more anomalies occurring within that time frame and 2) possible known and unknown cyber threats.

[080] The cyber threat analyst module 120 allows two levels of investigations of a cyber threat that may suggest a potential impending cyberattack. In a first level of investigation, the analyzer module 115 and AI model(s) 160 can rapidly detect and then the autonomous response engine 140 will autonomously respond to overt and obvious cyberattacks (generally indicated by high scores of 80 or more see Figure 3). However, thousands to millions of low level anomalies occur in a domain under analysis all of the time; and thus, most other systems need to set the threshold of trying to detect a cyberattack by a cyber threat at level higher such as a score of 80 or more than the low level anomalies examined by the cyber threat analyst module 120 just to not have too many false positive indications of a cyberattack when one is not actually occurring, as well as to not overwhelm a human cyber security analyst receiving the alerts with so many notifications of low level anomalies that they just start tuning out those alerts. However, advanced persistent threats attempt to avoid detection by making these low-level anomalies in the system over time during their cyberattack before making their final coup de grâce / ultimate mortal blow against the system (e.g., domain) being protected. The cyber threat analyst module 120 also conducts a second level of investigation over time with the assistance of the AI model(s) 160 trained with machine learning on how to form cyber threat hypotheses and how to conduct investigations for a cyber threat hypothesis that can detect these advanced persistent cyber threats actively trying to avoid detection by looking at one or more of these low-level anomalies combined in with other anomalies and factors as a part of a chain of linked information (See Figure 3).

[081] The cyber threat analyst module 120 forms in conjunction with the AI model(s) 160 trained with machine learning on how to form cyber threat hypotheses and how to conduct investigations for a cyber threat hypothesis investigate hypotheses on what are a possible set of cyber threats. The cyber threat analyst module 120 can also cooperate with the analyzer module 115 with its one or more data analysis processes to conduct an investigation on a possible set of cyber threats hypotheses that would include an anomaly of at least one of i) the abnormal behavior, ii) the suspicious activity, and iii) any

combination of both, identified through cooperation with, for example, the AI model(s) 160 trained with machine learning on the normal pattern of life of entities in the system. For example, as shown in Figure 3, the cyber threat analyst module 120 may perform several additional rounds 220 of gathering additional information, including abnormal behavior, over a period of time, in this example, examining data over a 7-day period to determine causal links between the information. The cyber threat analyst module 120 may submit to check and recheck various combinations / a chain of potentially related information, including abnormal behavior of a device/user account under analysis for example, until each of the one or more hypotheses on potential cyber threats are one of 1) refuted, 2) supported, or 3) included in a report that includes details of activities assessed to be relevant activities to the anomaly of interest to the user and that also conveys at least this particular hypothesis was neither supported or refuted. For this embodiment, a human cyber security analyst is then needed to further investigate the anomaly (and/or anomalies) of interest included in the chain of potentially related information.

[082] Returning back to Figure 4, an input from the cyber threat analyst module 120 of a supported hypothesis of a potential cyber threat will trigger the analyzer module 115 and/or assessment module 125 to compare, confirm, and send a signal to act upon and mitigate that cyber threat. In contrast, the cyber threat analyst module 120 investigates subtle indicators and/or initially seemingly isolated unusual or suspicious activity such as a worker is logging in after their normal working hours or a simple system misconfiguration has occurred. Most of the investigations conducted by the cyber threat analyst module 120 cooperating with the AI model(s) 160 trained with machine learning on how to form cyber threat hypotheses and how to conduct investigations for a cyber threat hypothesis on unusual or suspicious activities/behavior may not result in a cyber threat hypothesis that is supported but rather most cyber threat hypotheses are refuted or simply not supported. Typically, during the investigations, several rounds of data gathering to support or refute the long list of potential cyber threat hypotheses formed by the cyber threat analyst module 120 will occur before the algorithms in the cyber threat analyst module 120 will determine whether a particular cyber threat hypothesis is supported, refuted, or needs further investigation by a human. The rounds of data gathering will build chains of linked low-level indicators of unusual activity along with potential activities that

could be within a normal pattern of life for that entity to evaluate the whole chain of activities to support or refute each potential cyber threat hypothesis formed. (See again, for example, Figure 3 and a chain of linked low-level indicators, including abnormal behavior compared to the normal patten of life for that entity, all under a score of 50 on a threat indicator score). The investigations by the cyber threat analyst module 120 can happen over a relatively long period of time (e.g. a week or longer) and be far more in depth than the analyzer module 115 which will work with the other modules and AI model(s) 160 to confirm that a cyber threat has in fact been detected by the presence of an anomaly with a score of 75 or more and/or the occurrence of a specific event deemed a serious cyber threat in itself occurring.

[083] The gather module 110 cooperates with the cyber threat analyst module 120 and/or analyzer module 115 to collect data to support or to refute each of the one or more possible cyber threat hypotheses that could include this abnormal behavior or suspicious activity by cooperating with one or more of the cyber threat hypotheses mechanisms to form and investigate hypotheses on what are a possible set of cyber threats.

[084] Thus, the cyber threat analyst module 120 is configured to cooperate with the AI model(s) 160 trained with machine learning on how to form cyber threat hypotheses and how to conduct investigations for a cyber threat hypothesis to form and investigate hypotheses on what are a possible set of cyber threats and then can cooperate with the analyzer module 115 with the one or more data analysis processes to confirm the results of the investigation on the possible set of cyber threats hypotheses that would include the at least one of i) the abnormal behavior, ii) the suspicious activity, and iii) any combination of both, identified through cooperation with the AI model(s) 160 trained with machine learning on the normal pattern of life/normal behavior of entities in the domains under analysis.

[085] Note, in the first level of threat detection, the gather module 110 and the analyzer module 115 cooperate to supply any data and/or metrics requested by the analyzer module 115 cooperating with the AI model(s) 160 trained on possible cyber threats to support or rebut each possible type of cyber threat and generally that presence of an anomaly with a high threat/ anomaly score and/or the occurrence of a specific event

deemed a serious cyber threat in itself, will cause the analyzer module 115 to send a signal and this information to the autonomous response engine 140. Again, the analyzer module 115 can cooperate with the AI model(s) 160 and/or other modules to rapidly detect and then cooperate with the autonomous response engine 140 to autonomously respond to overt and obvious cyberattacks, (including ones found to be supported by the cyber threat analyst module 120).

[086] As a starting point, the AI-based cyber security appliance 100 can use multiple modules, each capable of identifying abnormal behavior and/or suspicious activity against the AI model(s) 160 trained on a normal pattern of life for the entities in the network/domain under analysis, which is supplied to the analyzer module 115 and/or the cyber threat analyst module 120. The analyzer module 115 and/or the cyber threat analyst module 120 may also receive other inputs such as AI model breaches, AI classifier breaches, etc. a trigger to start an investigation from an external source.

[087] Many other model breaches of the AI model(s) 160 trained with machine learning on the normal behavior of the system can send an input into the cyber threat analyst module 120 and/or the trigger module to trigger an investigation to start the formation of one or more hypotheses on what are a possible set of cyber threats that could include the initially identified abnormal behavior and/or suspicious activity.

[088] The cyber threat analyst module 120, which forms and investigates hypotheses on what are the possible set of cyber threats, can use hypotheses mechanisms including any of 1) one or more of the AI model(s) 160 trained on how human cyber security analysts form cyber threat hypotheses and how to conduct investigations for a cyber threat hypothesis that would include at least an anomaly of interest, 2) one or more scripts outlining how to conduct an investigation on a possible set of cyber threats hypotheses that would include at least the anomaly of interest, 3) one or more rules-based models on how to conduct an investigation on a possible set of cyber threats hypotheses and how to form a possible set of cyber threats hypotheses that would include at least the anomaly of interest, and 4) any combination of these. Again, the AI model(s) 160 trained on 'how to form cyber threat hypotheses and how to conduct investigations for a cyber threat hypothesis' may use supervised machine

learning on human-led cyber threat investigations and then steps, data, metrics, and metadata on how to support or to refute a plurality of the possible cyber threat hypotheses, and then the scripts and rules-based models will include the steps, data, metrics, and metadata on how to support or to refute the plurality of the possible cyber threat hypotheses. The cyber threat analyst module 120 and/or the analyzer module 115 can feed the cyber threat details to the assessment module 125 to generate a threat risk score that indicate a level of severity of the cyber threat.

[089] Each Artificial Intelligence-based engine has an interface to communicate with another separate Artificial Intelligence-based engine, which is configured to understand a type of information and communication that this other separate Artificial Intelligence-based engine needs to make determinations on an ongoing cyberattack from that other Artificial Intelligence-based engine's perspective. The autonomous response engine 140 works with the assessment module 125 in the detection engine when the cyber threat is detected and autonomously takes one or more actions to mitigate the cyber threat. Figure 4 shows the example components making up the detection engine to include interfaces to the cyber-attack prediction engine 105, the autonomous response engine 140, and the restoration engine 190.

[090] The cyber threat detection engine can also have an anomaly alert system in a formatting module configured to report out anomalous incidents and events as well as the cyber threat detected to a display screen viewable by a human cyber-security professional. Each Artificial Intelligence-based engine has a rapid messaging system to communicate with a human cyber-security team to keep the human cyber-security team informed on actions autonomously taken and actions needing human approval to be taken.

[091] Figure 5 illustrates a diagram of an embodiment of i) the cyber threat detection engine 100 using Artificial Intelligence algorithms trained to perform a first machine-learned task of detecting the cyber threat, ii) an autonomous response engine 140 using Artificial Intelligence algorithms trained to perform a second machine-learned task of taking one or more mitigation actions to mitigate the cyber threat, iii) a cyber-security restoration engine 190 using Artificial Intelligence algorithms trained to perform a third machine-learned task of remediating the system being protected back to a

trusted operational state, and iv) a cyber-attack prediction engine 105 using Artificial Intelligence algorithms trained to perform a fourth machine-learned task of Artificial Intelligence-based simulations of cyberattacks to assist in determining 1) how a simulated cyberattack might occur in the system being protected, and 2) how to use the simulated cyberattack information to preempt possible escalations of an ongoing actual cyberattack, in order for these four Artificial Intelligence-based engines to work together. In addition, the intelligent orchestration component can use Artificial Intelligence algorithms trained to perform a fifth machine-learned task of adaptive interactive response between the multiple Artificial Intelligence-based engines to provide information each Artificial Intelligence engine needs to work cohesively to provide an overall incidence response that mitigates different types of cyber threats while still minimizing an impact tailored to this particular system being protected. For example, when a conversation occurs between the AI-based engines such as a system that can be positively affected by both proposed mitigation actions and proposed restoration actions, any of which might be attempted but fail or only partially succeed, then the intelligent orchestration component can arbitrate and evolve the best result for this particular system being protected. The intelligent orchestration component can help anticipate i) the needs of and ii) cohesive response of each Artificial Intelligence-based engine based on a current detected cyber threat.

[092] Referring to Figure 5, the cyber security restoration engine 190 is configured to take one or more remediation actions with Artificial Intelligence assistance to remediate the one or more nodes in the graph of the system affected by the cyberattack back to a trusted operational state in a recovery from the cyber threat. These actions might be fully automatic, or require a specific human confirmation decision before they begin. The cyber security restoration engine 190 can cooperate with the other AI-based engines of the cyber security system, via the interfaces and/or direct integrations, to track and understand the cyber threat identified by the other components as well as track the one or more mitigation actions taken to mitigate the cyber threat during the cyberattack by the other components in order to assist in intelligently restoring the protected system while still mitigating the cyber threat attack back to a trusted operational state; and thus, as a situation develops with an ongoing

cyberattack, the cyber security restoration engine 190 is configured to take one or more remediation actions to remediate (e.g. restore) at least one of the nodes in the graph of the network back to a trusted operational state while the cyberattack is still ongoing.

[093] The example multiple Artificial Intelligence-based engines cooperating with each other can include i) the cyber threat detection engine, ii) an autonomous response engine 140, iii) a cyber-security restoration engine 190, and iv) a cyber-attack prediction engine 105. i) The cyber threat detection engine (consisting of the modules making up the cyber security appliance 100) can be configured to use Artificial Intelligence algorithms trained to perform a machine-learned task of detecting the cyber threat. (See for example Figure 4) ii) The autonomous response engine 140 can be configured to use Artificial Intelligence algorithms trained to perform a machine-learned task of taking one or more mitigation actions to mitigate, including stopping, the cyber threat. iii) The cyber-security restoration engine 190 can be configured to use Artificial Intelligence algorithms trained to perform a machine-learned task of remediating the system being protected back to a trusted operational state. iv) The cyber-attack prediction engine 105 can be configured to use Artificial Intelligence algorithms trained to perform a machine-learned task of Artificial Intelligence-based simulations of cyberattacks to assist in determining 1) how a simulated cyberattack might occur in the system being protected, and 2) how to use the simulated cyberattack information to preempt possible escalations of an ongoing actual cyberattack. (See, for example, Figure 6)

[094] The multiple Artificial Intelligence-based engines have communication hooks in between them to exchange a significant amount of behavioral metrics including data between the multiple Artificial Intelligence-based engines to work in together to provide an overall cyber threat response.

[095] The intelligent orchestration component can be configured as a discreet intelligent orchestration component that exists on top of the multiple Artificial Intelligence-based engines to orchestrate the overall cyber threat response and an interaction between the multiple Artificial Intelligence-based engines, each configured to perform its own machine-learned task. Alternatively, the intelligent orchestration component can be configured as a distributed collaboration with a portion of the

intelligent orchestration component implemented in each of the multiple Artificial Intelligence-based engines to orchestrate the overall cyber threat response and an interaction between the multiple Artificial Intelligence-based engines. In an embodiment, whether implemented as a distributed portion on each AI engine or a discrete AI engine itself, the intelligent orchestration component can use self-learning algorithms to learn how to best assist the orchestration of the interaction between itself and the other AI engines, which also implement self-learning algorithms themselves to perform their individual machine-learned tasks better.

[096] The multiple Artificial Intelligence-based engines can be configured to cooperate to combine an understanding of normal operations of the nodes, an understanding emerging cyber threats, an ability to contain those emerging cyber threats, and a restoration of the nodes of the system to heal the system with an adaptive feedback between the multiple Artificial Intelligence-based engines in light of simulations of the cyberattack to predict what might occur in the nodes in the system based on the progression of the attack so far, mitigation actions taken to contain those emerging cyber threats and remediation actions taken to heal the nodes using the simulated cyberattack information.

[097] The multiple Artificial Intelligence-based engines each have an interface to communicate with the other separate Artificial Intelligence-based engines configured to understand a type of information and communication that the other separate Artificial Intelligence-based engine needs to make determinations on an ongoing cyberattack from that other Artificial Intelligence-based engine's perspective. Each Artificial Intelligence-based engine has an instant messaging system to communicate with a human cyber-security team to keep the human cyber-security team informed on actions autonomously taken and actions needing human approval as well as generate reports for the human cyber-security team.

[098] Each of these Artificial Intelligence-based engines has bi-directional communications, including the exchange of raw data, with each other as well as with software agents resident in physical and /or virtual devices making up the system being protected as well as bi-directional communications with sensors within the system being protected. Note, the system under protection can be, for example, an IT network, an

OT network, a Cloud network, an email network, a source code database, an endpoint device, etc.

[099] In an example, the autonomous response engine 140 uses its intelligence to cooperate with a cyber-attack prediction engine and its Artificial Intelligence-based simulations to choose and initiate an initial set of one or more mitigation actions indicated as a preferred targeted initial response to the detected cyber threat by autonomously initiating those mitigation actions to defend against the detected cyber threat, rather than a human taking an action. The autonomous response engine 140, rather than the human taking the action, is configured to autonomously cause the one or more mitigation actions to be taken to contain the cyber threat when a threat risk parameter from an assessment module in the detection engine is equal to or above an actionable threshold. Example mitigation actions can include 1) the autonomous response engine 140 monitoring and sending signals to a potentially compromised node to restrict communications of the potentially compromised node to merely normal recipients and types of communications according to the Artificial Intelligence model trained to model the normal pattern of life for each node in the protected system, 2) the autonomous response engine 140 trained on how to isolate a compromised node as well as to take mitigation acts with other nodes that have a direct nexus to the compromised node.

[0100] In another example, the cyber-attack prediction engine 105 and its Artificial Intelligence-based simulations use intelligence to cooperate with the cyber-security restoration engine 190 to assist in choosing one or more remediation actions to perform on nodes affected by the cyberattack back to a trusted operational state while still mitigating the cyber threat during an ongoing cyberattack based on effects determined through the simulation of possible remediation actions to perform and their effects on the nodes making up the system being protected and preempt possible escalations of the cyberattack while restoring one or more nodes back to a trusted operational state.

[0101] In another example, the cyber security restoration engine 190 restores the one or more nodes in the protected system by cooperating with at least two or more of 1) an Artificial Intelligence model trained to model a normal pattern of life for each node

in the protected system, 2) an Artificial Intelligence model trained on what are a possible set of cyber threats and their characteristics and symptoms to identify the cyber threat (e.g. malicious actor/device/file) that is causing a particular node to behave abnormally (e.g. malicious behavior) and fall outside of that node's normal pattern of life, and 3) the autonomous response engine 140.

[0102] Figure 6 illustrates a block diagram of an embodiment of the cyber-attack prediction engine with Artificial Intelligence-based simulations conducted in the cyber-attack prediction engine by constructing a graph of nodes of the system being protected (e.g. a network) including i) the physical devices connecting to the network, any virtualized instances of the network, user accounts in the network, email accounts in the network, etc. as well as ii) connections and pathways through the network to create a virtualized instance of the network to be tested. As shown in Figure 6, the various cooperating modules residing in the cyber-attack prediction engine 105 may include, but are not limited to, a collections module 705, a cyberattack generator (e.g. phishing email generator with a paraphrasing engine) 702, an email module 715, a network module 720, an analyzer module 725, a payloads module 730 with first and second payloads, a communication module 735, a training module 740, a simulated attack module 750, a cleanup module 755, a scenario module 760, a user interface 765, a reporting module, a formatting module, an orchestration module, an AI classifier with a list of specified classifiers.

[0103] The cyber-attack prediction engine 105 may be implemented via i) a simulator to model the system being protected and/or ii) a clone creator to spin up a virtual network and create a virtual clone of the system being protected configured to pentest one or more defenses provided by scores based on both the level of confidence that the cyber threat is a viable threat and the severity of the cyber threat (e.g., attack type where ransomware attacks has greater severity than phishing attack; degree of infection; computing devices likely to be targeted, etc.). The threat risk scores be used to rank alerts that may be directed to enterprise or computing device administrators. This risk assessment and ranking is conducted to avoid frequent "false positive" alerts that diminish the degree of reliance/confidence on the cyber security appliance 100. The cyber-attack prediction engine 105 may include and cooperate with one or more AI

models trained with machine learning on the contextual knowledge of the organization. These trained AI models may be configured to identify data points from the contextual knowledge of the organization and its entities, which may include, but is not limited to, language-based data, email/network connectivity and behavior pattern data, and/or historic knowledgebase data. The cyber-attack prediction engine 105 may use the trained AI models to cooperate with one or more AI classifier(s) by producing a list of specific organization-based classifiers for the AI classifier. The cyber-attack prediction engine 105 is further configured to calculate,-based at least in part on the results of the one or more hypothetical simulations of a possible cyberattack path and/or of an actual ongoing cyberattack paths from a cyber threat determine a risk score for each node (e.g. each device, user account, etc.), the threat risk score being indicative of a possible severity of the compromise prior to an autonomous response action is taken in response to the actual cyberattack of the cyber incident. See for example Figures 7A and 7B.

[0104] Figure 7A illustrates a diagram of an embodiment of the cyber-attack prediction engine and its Artificial Intelligence-based simulations constructing an example graph of nodes in an example network and simulating how the cyberattack path might likely progress in the future tailored with an innate understanding of a normal behavior of the nodes in the system being protected and a current operational state of each node in the graph of the protected system during simulations of cyberattacks. The cyber-attack prediction engine 105 plots the attack path through the nodes and estimated times to reach critical nodes in the network. The cyberattack simulation modeling is run to identify the routes, difficulty, and time periods from certain entry notes to certain key servers.

[0105] Again, similarly named components in each Artificial Intelligence-based engine can 1) perform similar functions and/or 2) have a communication link from that component located in one of the Artificial Intelligence-based engines and then information is needed from that component is communicated to another Artificial Intelligence-based engine that through the interface to that Artificial Intelligence-based engine.

[0106] Figure 7B illustrates a diagram of an embodiment of the cyber-attack prediction engine and/or the cyber-attack restoration engine assigning scores for a portion of the graph of nodes of the system being protected (e.g. a network) including i) the physical devices, accounts, etc. in the system, etc. as well as ii) connections and attack pathways through the network.

[0107] Training of AI pre-deployment and then during deployment

[0108] In step 1, an initial training of the Artificial Intelligence model trained on cyber threats can occur using unsupervised learning and/or supervised learning on characteristics and attributes of known potential cyber threats including malware, insider threats, and other kinds of cyber threats that can occur within that domain. Each Artificial Intelligence model (e.g. neural network, decision tree, etc.) can be programmed and configured with the background information to understand and handle particulars, including different types of data, protocols used, types of devices, user accounts, etc. of the system being protected. The Artificial Intelligence pre-deployment can all be trained on the specific machine learning task that they will perform when put into deployment. For example, the AI model, such as AI model(s) 160 or example (hereinafter "AI model(s) 160"), trained on identifying a specific cyber threat learns at least both in the pre-deployment training i) the characteristics and attributes of known potential cyber threats as well as ii) a set of characteristics and attributes of each category of potential cyber threats and their weights assigned on how indicative certain characteristics and attributes correlate to potential cyber threats of that category of threats. In this example, one of the AI models 160 trained on identifying a specific cyber threat can be trained with machine learning such as Linear Regression, Regression Trees, Non-Linear Regression, Bayesian Linear Regression, Deep learning, etc. to learn and understand the characteristics and attributes in that category of cyber threats. Later, when in deployment in a domain/network being protected by the cyber security appliance 100, the AI model trained on cyber threats can determine whether a potentially unknown threat has been detected via a number of techniques including an overlap of some of the same characteristics and attributes in that category of cyber threats. The AI model may use unsupervised learning when deployed to better learn newer and updated characteristics of cyberattacks.

[0109] In an embodiment, one or more of the AI models 160 may be trained on a normal pattern of life of entities in the system are self-learning AI model using unsupervised machine learning and machine learning algorithms to analyze patterns and 'learn' what is the 'normal behavior' of the network by analyzing data on the activity on, for example, the network level, at the device level, and at the employee level. The self-learning AI model using unsupervised machine learning understands the system under analysis' normal patterns of life in, for example, a week of being deployed on that system, and grows more bespoke with every passing minute. The AI unsupervised learning model learns patterns from the features in the day-to-day dataset and detecting abnormal data which would not have fallen into the category (cluster) of normal behavior. The self-learning AI model using unsupervised machine learning can simply be placed into an observation mode for an initial week or two when first deployed on a network/domain in order to establish an initial normal behavior for entities in the network/domain under analysis.

[0110] Thus, a deployed Artificial Intelligence model 160 trained on a normal behavior of entities in the system can be configured to observe the nodes in the system being protected. Training on a normal behavior of entities in the system can occur while monitoring for the first week or two until enough data has been observed to establish a statistically reliable set of normal operations for each node (e.g., user account, device, etc.). Initial training of one or more Artificial Intelligence models 160 trained with machine learning on a normal behavior of the pattern of life of the entities in the network/domain can occur where each type of network and/or domain will generally have some common typical behavior with each model trained specifically to understand components/devices, protocols, activity level, etc. to that type of network/system/domain. Alternatively, pre-deployment machine learning training of one or more Artificial Intelligence models trained on a normal pattern of life of entities in the system can occur. Initial training of one or more Artificial Intelligence models trained with machine learning on a normal behavior of the pattern of life of the entities in the network/domain can occur where each type of network and/or domain will generally have some common typical behavior with each model trained specifically to understand components/devices, protocols, activity level, etc. to that type of

network/system/domain. What is the normal behavior of each entity within that system can be established either prior to the deployment and then adjusted during deployment or alternatively the model can simply be placed into an observation mode for an initial week or two when first deployed on a network/domain in order to establish an initial normal behavior for entities in the network/domain under analysis. During the deployment of the model, what is considered normal behavior will change as each different entity's behavior changes and will be reflected through the use of unsupervised learning in the model such as various Bayesian techniques, clustering, etc. Again, the AI models 160 can be implemented with various mechanisms, such neural networks, decision trees, etc. and combinations of these. Likewise, one or more supervised machine learning AI models 160 may be trained to create possible hypotheses and perform cyber threat investigations on agnostic examples of past historical incidents of detecting a multitude of possible types of cyber threat hypotheses previously analyzed by human cyber security analyst.

[0111] At its core, the self-learning AI models 160 that model the normal behavior (e.g. a normal pattern of life) of entities in the network mathematically characterizes what constitutes 'normal' behavior, based on the analysis of a large number of different measures of a device's network behavior - packet traffic and network activity/processes including server access, data volumes, timings of events, credential use, connection type, volume, and directionality of, for example, uploads/downloads into the network, file type, packet intention, admin activity, resource and information requests, command sent, etc.

[0112] Clustering Methods

[0113] In order to model what should be considered as normal for a device or cloud container, its behavior can be analyzed in the context of other similar entities on the network. The AI models (e.g., AI model(s) 160) can use unsupervised machine learning to algorithmically identify significant groupings, a task which is virtually impossible to do manually. To create a holistic image of the relationships within the network, the AI models and AI classifiers employ a number of different clustering methods, including matrix-based clustering, density-based clustering, and hierarchical

clustering techniques. The resulting clusters can then be used, for example, to inform the modeling of the normative behaviors and/or similar groupings.

[0114] The AI models and AI classifiers can employ a large-scale computational approach to understand sparse structure in models of network connectivity based on applying L1- regularization techniques (the lasso method). This allows the artificial intelligence to discover true associations between different elements of a network which can be cast as efficiently solvable convex optimization problems and yield parsimonious models. Various mathematical approaches assist.

[0115] Next, one or more supervised machine learning AI models are trained to create possible hypotheses and how to perform cyber threat investigations on agnostic examples of past historical incidents of detecting a multitude of possible types of cyber threat hypotheses previously analyzed by human cyber threat analysis. AI models 160 trained on forming and investigating hypotheses on what are a possible set of cyber threats can be trained initially with supervised learning. Thus, these AI models 160 can be trained on how to form and investigate hypotheses on what are a possible set of cyber threats and steps to take in supporting or refuting hypotheses. The AI models trained on forming and investigating hypotheses are updated with unsupervised machine learning algorithms when correctly supporting or refuting the hypotheses including what additional collected data proved to be the most useful. More on the training of the AI models that are trained to create one or more possible hypotheses and perform cyber threat investigations will be discussed later.

[0116] Next, the various Artificial Intelligence models and AI classifiers combine use of unsupervised and supervised machine learning to learn 'on the job' – it does not depend upon solely knowledge of previous cyber threat attacks. The Artificial Intelligence models and classifiers combine use of unsupervised and supervised machine learning constantly revises assumptions about behavior, using probabilistic mathematics, that is always up to date on what a current normal behavior is, and not solely reliant on human input. The Artificial Intelligence models and classifiers combine use of unsupervised and supervised machine learning on cyber security is capable of seeing hitherto undiscovered cyber events, from a variety of threat sources, which would otherwise have gone unnoticed. Next, these cyber threats can include, for

example: Insider threat – malicious or accidental, Zero-day attacks – previously unseen, novel exploits, latent vulnerabilities, machine-speed attacks – ransomware and other automated attacks that propagate and/or mutate very quickly, Cloud and SaaS-based attacks, other silent and stealthy attacks advance persistent threats, advanced spear-phishing, etc.

[0117] Ranking the Cyber Threat

[0118] The assessment module 125 and/or cyber threat analyst module 120 of Figure 4 can cooperate with the AI model(s) 160 trained on possible cyber threats to use AI algorithms to account for ambiguities by distinguishing between the subtly differing levels of evidence that characterize network data. Instead of generating the simple binary outputs 'malicious' or 'benign', the AI's mathematical algorithms produce outputs marked with differing degrees of potential threat. This enables users of the system to rank alerts and notifications to the enterprise security administrator in a rigorous manner, and prioritize those which most urgently require action. Meanwhile, it also assists to avoid the problem of numerous false positives associated with simply a rule-based approach.

[0119] More on the operation of the cyber security appliance

[0120] As discussed in more detail below, the analyzer module 115 and/or cyber threat analyst module 120 can cooperate with the one or more unsupervised AI (machine learning) model 160 trained on the normal pattern of life/normal behavior in order to perform anomaly detection against the actual normal pattern of life for that system to determine whether an anomaly (e.g., the identified abnormal behavior and/or suspicious activity) is malicious or benign. In the operation of the cyber security appliance 100, the emerging cyber threat can be previously unknown, but the emerging threat landscape data 170 representative of the emerging cyber threat shares enough (or does not share enough) in common with the traits from the AI models 160 trained on cyber threats to now be identified as malicious or benign. Note, if later confirmed as malicious, then the AI models 160 trained with machine learning on possible cyber threats can update their training. Likewise, as the cyber security appliance 100 continues to operate, then the one or more AI models trained on a normal pattern of life for each of the entities in the system can be updated and trained with unsupervised

machine learning algorithms. The analyzer module 115 can use any number of data analysis processes (discussed more in detail below and including the agent analyzer data analysis process here) to help obtain system data points so that this data can be fed and compared to the one or more AI models trained on a normal pattern of life, as well as the one or more machine learning models trained on potential cyber threats, as well as create and store data points with the connection fingerprints.

[0121] All of the above AI models 160 can continually learn and train with unsupervised machine learning algorithms on an ongoing basis when deployed in their system that the cyber security appliance 100 is protecting. Thus, learning and training on what is normal behavior for each user, each device, and the system overall and lowering a threshold of what is an anomaly.

[0122] Anomaly detection/ deviations

[0123] Anomaly detection can discover unusual data points in your dataset. Anomaly can be a synonym for the word 'outlier'. Anomaly detection (or outlier detection) is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. Anomalous activities can be linked to some kind of problems or rare events. Since there are tons of ways to induce a particular cyber-attack, it is very difficult to have information about all these attacks beforehand in a dataset. But, since the majority of the user activity and device activity in the system under analysis is normal, the system overtime captures almost all of the ways which indicate normal behavior. And from the inclusion-exclusion principle, if an activity under scrutiny does not give indications of normal activity, the self-learning AI model using unsupervised machine learning can predict with high confidence that the given activity is anomalous/unusual. The AI unsupervised learning model learns patterns from the features in the day to day dataset and detecting abnormal data which would not have fallen into the category (cluster) of normal behavior. The goal of the anomaly detection algorithm through the data fed to it is to learn the patterns of a normal activity so that when an anomalous activity occurs, the modules can flag the anomalies through the inclusion-exclusion principle. The goal of the anomaly detection algorithm through the data fed to it is to learn the patterns of a normal activity so that when an anomalous activity occurs, the modules can flag the anomalies through the

inclusion-exclusion principle. The cyber threat module can perform its two level analysis on anomalous behavior and determine correlations.

[0124] In an example, 95% of data in a normal distribution lies within two standard-deviations from the mean. Since the likelihood of anomalies in general is very low, the modules cooperating with the AI model of normal behavior can say with high confidence that data points spread near the mean value are non-anomalous. And since the probability distribution values between mean and two standard-deviations are large enough, the modules cooperating with the AI model of normal behavior can set a value in this example range as a threshold (a parameter that can be tuned over time through the self-learning), where feature values with probability larger than this threshold indicate that the given feature's values are non-anomalous, otherwise it's anomalous. Note, this anomaly detection can determine that a data point is anomalous/non-anomalous on the basis of a particular feature. In reality, the cyber security appliance 100 should not flag a data point as an anomaly based on a single feature. Merely, when a combination of all the probability values for all features for a given data point is calculated can the modules cooperating with the AI model of normal behavior can say with high confidence whether a data point is an anomaly or not. Anomaly detection can discover unusual data points in your dataset.

[0125] Again, the AI models trained on a normal pattern of life of entities in a network (e.g., domain) under analysis may perform the cyber threat detection through a probabilistic change in a normal behavior through the application of, for example, an unsupervised Bayesian mathematical model to detect the behavioral change in computers and computer networks. The Bayesian probabilistic approach can determine periodicity in multiple time series data and identify changes across single and multiple time series data for the purpose of anomalous behavior detection. Please reference US patent 10,701,093 granted June 30th, 2020, titled "Anomaly alert system for cyber threat detection" for an example Bayesian probabilistic approach, which is incorporated by reference in its entirety. In addition, please reference US patent publication number "US2021273958A1 filed February 26, 2021, titled "Multi-stage anomaly detection for process chains in multi-host environments" for another example anomalous behavior detector using a recurrent neural network and a bidirectional long short-term memory

(LSTM), which is incorporated by reference in its entirety. In addition, please reference US patent publication number "US2020244673A1, filed April 23, 2019, titled "Multivariate network structure anomaly detector," which is incorporated by reference in its entirety, for another example anomalous behavior detector with a Multivariate Network and Artificial Intelligence classifiers.

[0126] Next, as discussed further below, as discussed further below, during pre-deployment the cyber threat analyst module 120 and the analyzer module 115 can use data analysis processes and cooperate with AI model(s) 160 trained on forming and investigating hypotheses on what are a possible set of cyber threats. In addition, another set of AI models can be trained on how to form and investigate hypotheses on steps to take in supporting or refuting hypotheses. The AI models trained on forming and investigating hypotheses are updated with unsupervised machine learning algorithms when correctly supporting or refuting the hypotheses including what additional collected data proved to be the most useful.

[0127] **Additional module interactions**

[0128] Referring back to Figure 4, the gather module 110 cooperates with the data store 135. The data store 135 stores comprehensive logs for network traffic observed, email activity, cloud activity, etc. each domain can store their long term data storage in the data store. These logs can be filtered with complex logical queries and each, for example, IP packet can be interrogated on a vast number of metrics in the network information stored in the data store. The gather module 110 pulls data relevant for each possible hypothesis from the data store as well as from additional external and internal sources. In an example, the data store 135 can store the metrics and previous threat alerts associated with network traffic for a period of time, which is, by default, at least 27 days. This corpus of data is fully searchable. The cyber security appliance 100 works with network probes to monitor network traffic and store and record the data and metadata associated with the network traffic in the data store.

[0129] The gather module 110 may have a process identifier classifier. The process identifier classifier can identify and track each process and device in the network, under analysis, making communication connections. The data store 135 cooperates with the process identifier classifier to collect and maintain historical data of

processes and their connections, which is updated over time as the network is in operation. In an example, the process identifier classifier can identify each process running on a given device along with its endpoint connections, which are stored in the data store. Similarly, data from any of the domains under analysis may be collected and compared. Examples of domains/networks under analysis being protected can include any of i) an Informational Technology network, ii) an Operational Technology network, iii) a Cloud service, iv) a SaaS service, v) an endpoint device, vi) an email domain, and vii) any combinations of these.

[0130] A domain module is constructed and coded to interact with and understand a specific domain. For instance, the IT network domain module 145 may receive information from and send information to, in this example, IT network-based sensors (i.e., probes, taps, etc.). The IT network domain module 145 also has algorithms and components configured to understand, in this example, IT network parameters, IT network protocols, IT network activity, and other IT network characteristics of the network under analysis. The second domain module 150 is, in this example, an email module. The email domain module 150 can receive information from and send information to, in this example, email-based sensors (i.e., probes, taps, etc.). The email domain module 150 also has algorithms and components configured to understand, in this example, email parameters, email protocols and formats, email activity, and other email characteristics of the network under analysis. Additional domain modules, such as a cloud domain module can also collect domain data from another respective domain.

[0131] The coordinator module 155 is configured to work with various machine learning algorithms and relational mechanisms to i) assess, ii) annotate, and/or iii) position in a vector diagram, a directed graph, a relational database, etc., activity including events occurring, for example, in the first domain compared to activity including events occurring in the second domain. The domain modules can cooperate to exchange and store their information with the data store.

[0132] As discussed, the process identifier classifier in the gather module 110 can cooperate with additional classifiers in each of the domain modules 145/150 to assist in tracking individual processes and associating them with entities in a domain

under analysis as well as individual processes and how they relate to each other. The process identifier classifier can cooperate with other trained AI classifiers in the modules to supply useful metadata along with helping to make logical nexuses. A feedback loop of cooperation exists between the gather module 110, the analyzer module 115, the domain specific modules such as the IT network module and/or email module, the AI model(s) 160 trained on different aspects of this process, and the cyber threat analyst module 120 to gather information to determine whether a cyber threat is potentially attacking the networks/domains under analysis.

[0133] **Determination of whether something is likely malicious**

[0134] In the following examples the analyzer module 115 and/or cyber threat analyst module 120 can use multiple factors to the determination of whether a process, event, object, entity, etc. is likely malicious.

[0135] In an example, the analyzer module 115 and/or cyber threat analyst module 120 can cooperate with one or more of the AI model(s) 160 trained on certain cyber threats to detect whether the anomalous activity detected, such as suspicious email messages, exhibit traits that may suggest a malicious intent, such as phishing links, scam language, sent from suspicious domains, etc. The analyzer module 115 and/or cyber threat analyst module 120 can also cooperate with one of more of the AI model(s) 160 trained on potential IT based cyber threats to detect whether the anomalous activity detected, such as suspicious IT links, URLs, domains, user activity, etc., may suggest a malicious intent as indicated by the AI models trained on potential IT based cyber threats.

[0136] In the above example, the analyzer module 115 and/or the cyber threat analyst module 120 can cooperate with the one or more AI models 160 trained with machine learning on the normal pattern of life for entities in an email domain under analysis to detect, in this example, anomalous emails which are detected as outside of the usual pattern of life for each entity, such as a user, email server, etc., of the email network/domain. Likewise, the analyzer module 115 and/or the cyber threat analyst module 120 can cooperate with the one or more AI models trained with machine learning on the normal pattern of life for entities in a second domain under analysis (in this example, an IT network) to detect, in this example, anomalous network activity by

user and/or devices in the network, which is detected as outside of the usual pattern of life (e.g. abnormal) for each entity, such as a user or a device, of the second domain's network under analysis.

[0137] Thus, the analyzer module 115 and/or the cyber threat analyst module 120 can be configured with one or more data analysis processes to cooperate with the one or more of the AI model(s) 160 trained with machine learning on the normal pattern of life in the system, to identify an anomaly of at least one of i) the abnormal behavior, ii) the suspicious activity, and iii) the combination of both, from one or more entities in the system. Note, other sources, such as other model breaches, can also identify at least one of i) the abnormal behavior, ii) the suspicious activity, and iii) the combination of both to trigger the investigation.

[0138] Accordingly, during this cyber threat determination process, the analyzer module 115 and/or the cyber threat analyst module 120 can also use AI classifiers that look at the features and determine a potential maliciousness based on commonality or overlap with known characteristics of malicious processes/entities. Many factors, including anomalies that include unusual and suspicious behavior, and other indicators of processes and events, are examined by the one or more AI models 160 trained on potential cyber threats including some supporting AI classifiers looking at specific features for their malicious nature in order to make a determination of whether an individual factor and/or whether a chain of anomalies is determined to be likely malicious.

[0139] Initially, in this example of activity in an IT network analysis, the rare JA3 hash and/or rare user agent connections for this network coming from a new or unusual process are factored just like in the first wireless domain suspicious wireless signals are considered. These are quickly determined by referencing the one or more of the AI model(s) 160 trained with machine learning on the pattern of life of each device and its associated processes in the system. Next, the analyzer module 115 and/or the cyber threat analyst module 120 can have an external input to ingest threat intelligence from other devices in the network cooperating with the cyber security appliance 100. Next, the analyzer module 115 and/or the cyber threat analyst module 120 can look for other anomalies, such as model breaches, while the AI models trained on potential cyber

threats can assist in examining and factoring other anomalies that have occurred over a given timeframe to see if a correlation exists between a series of two or more anomalies occurring within that time frame.

[0140] The analyzer module 115 and/or the cyber threat analyst module 120 can combine these Indicators of Compromise (e.g., unusual network JA3, unusual device JA3, ...) with many other weak indicators to detect the earliest signs of an emerging threat, including previously unknown threats, without using strict blacklists or hard-coded thresholds. However, the AI classifiers can also routinely look at blacklists, etc. to identify maliciousness of features looked at. A deeper analysis may assist in confirming an analysis to determine that indeed a cyber threat has been detected. The analyzer module 115 can also look at factors of how rare the endpoint connection is, how old the endpoint is, where geographically the endpoint is located, how a security certificate associated with a communication is verified only by an endpoint device or by an external 3rd party, just to name a few additional factors. The analyzer module 115 (and similarly the cyber threat analyst module 120) can then assign weighting given to these factors in the machine learning that can be supervised based on how strongly that characteristic has been found to match up to actual malicious cyber threats learned in the training.

[0141] In another example, an AI classifier supporting the AI models 160 is trained to find potentially malicious indicators. The agent analyzer data analysis process in the analyzer module 115 and/or cyber threat analyst module 120 may cooperate with the process identifier classifier to identify all of the additional factors of i) are one or more processes running independently of other processes, ii) are the one or more processes running independent are recent to this network, and iii) are the one or more processes running independent connect to the endpoint, which the endpoint is a rare connection for this network, which are referenced and compared to one or more AI models 160 trained with machine learning on the normal behavior of the pattern of life of the system.

[0142] The analyzer module 115 and/or the cyber threat analyst module 120 may use the agent analyzer data analysis process that detects a potentially malicious agent previously unknown to the system to start an investigation on one or more possible

cyber threat hypotheses. The determination and output of this step is what are possible cyber threats that can include or be indicated by the identified abnormal behavior and/or identified suspicious activity identified by the agent analyzer data analysis process.

[0143] In an example, the cyber threat analyst module 120 can use the agent analyzer data analysis process and the AI models(s) trained on forming and investigating hypotheses on what are a possible set of cyber threats to use the machine learning and/or set scripts to aid in forming one or more hypotheses to support or refute each hypothesis. The cyber threat analyst module 120 can cooperate with the AI models trained on forming and investigating hypotheses to form an initial set of possible hypotheses, which needs to be intelligently filtered down. The cyber threat analyst module 120 can be configured to use the one or more supervised machine learning models trained on i) agnostic examples of a past history of detection of a multitude of possible types of cyber threat hypotheses previously analyzed by human, who was a cyber security professional, ii) a behavior and input of how a plurality of human cyber security analysts make a decision and analyze a risk level regarding and a probability of a potential cyber threat, iii) steps to take to conduct an investigation start with anomaly via learning how expert humans tackle investigations into specific real and synthesized cyber threats and then the steps taken by the human cyber security professional to narrow down and identify a potential cyber threat, and iv) what type of data and metrics that were helpful to further support or refute each of the types of cyber threats, in order to determine a likelihood of whether the abnormal behavior and/or suspicious activity is either i) malicious or ii) benign?

[0144] The cyber threat analyst module 120 using AI models, scripts and/or rules based modules is configured to conduct initial investigations regarding the anomaly of interest, collected additional information to form a chain of potentially related/linked information under analysis and then form one or more hypotheses that could have this chain of information that is potentially related/linked under analysis and then gather additional information in order to refute or support each of the one or more hypotheses.

[0145] The cyber threat analyst module using AI models, scripts and/or rules-based modules is configured to conduct initial investigations regarding the anomaly of interest, collected additional information to form a chain of potentially related/linked

information under analysis and then form one or more hypotheses that could have this chain of information that is potentially related/linked under analysis and then gather additional information in order to refute or support each of the one or more hypotheses.

[0146] In an example, a behavioural pattern analysis of what are the unusual behaviours of the network/system/device/user under analysis by the machine learning models may be as follows. The coordinator module can tie the alerts, activities, and events from, in this example, the email domain to the alerts, activities, and events from the IT network domain. Figure 3 illustrates a graph 220 of an embodiment of an example chain of unusual behaviour for, in this example, the email activities and IT network activities deviating from a normal pattern of life in connection with the rest of the system/network under analysis. The cyber threat analyst module and/or analyzer module can cooperate with one or more machine learning models. The one or more machine learning models are trained and otherwise configured with mathematical algorithms to infer, for the cyber-threat analysis, 'what is possibly happening with the chain of distinct alerts, activities, and/or events, which came from the unusual pattern,' and then assign a threat risk associated with that distinct item of the chain of alerts and/or events forming the unusual pattern. The unusual pattern can be determined by examining initially what activities/events/alerts that do not fall within the window of what is the normal pattern of life for that network/system/device/user under analysis can be analysed to determine whether that activity is unusual or suspicious. A chain of related activity that can include both unusual activity and activity within a pattern of normal life for that entity can be formed and checked against individual cyber threat hypothesis to determine whether that pattern is indicative of a behaviour of a malicious actor – human, program, or other threat. The cyber threat analyst module can go back and pull in some of the normal activities to help support or refute a possible hypothesis of whether that pattern is indicative of a behavior of a malicious actor. An example behavioral pattern included in the chain is shown in the graph over a time frame of, an example, 7 days. The cyber threat analyst module detects a chain of anomalous behavior of unusual data transfers three times, unusual characteristics in emails in the monitored system three times which seem to have some causal link to the unusual data transfers. Likewise, twice unusual credentials attempted the unusual behavior of trying

to gain access to sensitive areas or malicious IP addresses and the user associated with the unusual credentials trying unusual behavior has a causal link to at least one of those three emails with unusual characteristics. Again, the cyber security appliance 100 can go back and pull in some of the normal activities to help support or refute a possible hypothesis of whether that pattern is indicative of a behaviour of a malicious actor. The analyser module can cooperate with one or more models trained on cyber threats and their behaviour to try to determine if a potential cyber threat is causing these unusual behaviours. The cyber threat analyst module can put data and entities into 1) a directed graph and nodes in that graph that are overlapping or close in distance have a good possibility of being related in some manner, 2) a vector diagram, 3) a relational database, and 4) other relational techniques that will at least be examined to assist in creating the chain of related activity connected by causal links, such as similar time, similar entity and/or type of entity involved, similar activity, etc., under analysis. If the pattern of behaviours under analysis is believed to be indicative of a malicious actor, then a score of how confident is the system in this assessment of identifying whether the unusual pattern was caused by a malicious actor is created. Next, also assigned is a threat level score or probability indicative of what level of threat does this malicious actor pose. Lastly, the cyber security appliance 100 is configurable in a user interface, by a user, enabling what type of automatic response actions, if any, the cyber security appliance 100 may take when different types of cyber threats, indicated by the pattern of behaviours under analysis, that are equal to or above a configurable level of threat posed by this malicious actor. The chain of the individual alerts, activities, and events that form the pattern including one or more unusual or suspicious activities into a distinct item for cyber-threat analysis of that chain of distinct alerts, activities, and/or events. The cyber-threat module may reference the one or more machine learning models trained on, in this example, e-mail threats to identify similar characteristics from the individual alerts and/or events forming the distinct item made up of the chain of alerts and/or events forming the unusual pattern.

[0147] The autonomous response engine 140 of the cyber security system is configured to take one or more autonomous mitigation actions to mitigate the cyber threat during the cyberattack by the cyber threat. The autonomous response engine

140 is configured to reference an Artificial Intelligence model trained to track a normal pattern of life for each node of the protected system to perform an autonomous act of restricting a potentially compromised node having i) an actual indication of compromise and/or ii) merely adjacent to a known compromised node, to merely take actions that are within that node's normal pattern of life to mitigate the cyber threat. Similarly named components in the cyber security restoration engine 190 can operate and function similar to as described for the detection engine.

[0148] An assessment of the cyber threat in order to determine appropriate autonomous actions, for example, those by the autonomous response engine

[0149] In the next step, the analyzer module 115 and/or cyber threat analyst module 120 generates one or more supported possible cyber threat hypotheses from the possible set of cyber threat hypotheses. The analyzer module generates the supporting data and details of why each individual hypothesis is supported or not. The analyzer module can also generate one or more possible cyber threat hypotheses and the supporting data and details of why they were refuted.

[0150] In general, the analyzer module 115 cooperates with the following three sources. The analyzer module 115 cooperates with the AI models trained on cyber threats to determine whether an anomaly such as the abnormal behavior and/or suspicious activity is either 1) malicious or 2) benign when the potential cyber threat under analysis is previously unknown to the cyber security appliance 100. The analyzer module cooperates with the AI models trained on a normal behavior of entities in the network under analysis. The analyzer module cooperates with various AI-trained classifiers. With all of these sources, when they input information that indicates a potential cyber threat that is i) severe enough to cause real harm to the network under analysis and/or ii) a close match to known cyber threats, then the analyzer module can make a final determination to confirm that a cyber threat likely exists and send that cyber threat to the assessment module to assess the threat score associated with that cyber threat. Certain model breaches will always trigger a potential cyber threat that the analyzer will compare and confirm the cyber threat.

[0151] In the next step, an assessment module with the AI classifiers is configured to cooperate with the analyzer module. The analyzer module supplies the

identity of the supported possible cyber threat hypotheses from the possible set of cyber threat hypotheses to the assessment module. The assessment module with the AI classifiers cooperates with the AI model trained on possible cyber threats can make a determination on whether a cyber threat exists and what level of severity is associated with that cyber threat. The assessment module with the AI classifiers cooperates with the one or more AI models trained on possible cyber threats in order to assign a numerical assessment of a given cyber threat hypothesis that was found likely to be supported by the analyzer module with the one or more data analysis processes, via the abnormal behavior, the suspicious activity, or the collection of system data points. The assessment module with the AI classifiers output can be a score (ranked number system, probability, etc.) that a given identified process is likely a malicious process.

[0152] The assessment module with the AI classifiers can be configured to assign a numerical assessment, such as a probability, of a given cyber threat hypothesis that is supported and a threat level posed by that cyber threat hypothesis which was found likely to be supported by the analyzer module, which includes the abnormal behavior or suspicious activity as well as one or more of the collection of system data points, with the one or more AI models trained on possible cyber threats.

[0153] The cyber threat analyst module 120 in the AI-based cyber security appliance 100 component provides an advantage over competitors' products as it reduces the time taken for cybersecurity investigations, provides an alternative to manpower for small organizations and improves detection (and remediation) capabilities within the cyber security platform.

[0154] The AI-based cyber threat analyst module 120 performs its own computation of threat and identifies interesting network events with one or more processers. These methods of detection and identification of threat all add to the above capabilities that make the AI-based cyber threat analyst module a desirable part of the cyber security appliance 100. The AI-based cyber threat analyst module 120 offers a method of prioritizing which is not just a summary or highest score alert of an event evaluated by itself equals the most bad, and prevents more complex attacks being missed because their composite parts/individual threats only produced low-level alerts.

[0155] The AI classifiers can be part of the assessment component, which scores the outputs of the analyzer module. Again, as for the other AI classifiers discussed, the AI classifier can be coded to take in multiple pieces of information about an entity, object, and/or thing and based on its training and then output a prediction about the entity, object, or thing. Given one or more inputs, the AI classifier model will try to predict the value of one or more outcomes. The AI classifiers cooperate with the range of data analysis processes that produce features for the AI classifiers. The various techniques cooperating here allow anomaly detection and assessment of a cyber threat level posed by a given anomaly; but more importantly, an overall cyber threat level posed by a series/chain of correlated anomalies under analysis.

[0156] In the next step, the formatting module can generate an output such as a printed or electronic report with the relevant data. The formatting module can cooperate with both the analyzer module, the cyber threat analyst module, and the assessment module depending on what the user wants to be reported.

[0157] The formatting module is configured to format, present a rank for, and output one or more detected cyber threats from the analyzer module or from the assessment module into a formalized report, from one or more report templates populated with the data for that incident. Many different types of formalized report templates exist to be populated with data and can be outputted in an easily understandable format for a human user's consumption.

[0158] The formalized report on the template is outputted for a human user's consumption in a medium of any of 1) printable report, 2) presented digitally on a user interface, 3) in a machine readable format for further use in machine-learning reinforcement and refinement, or 4) any combination of the three. The formatting module is further configured to generate a textual write up of an incident report in the formalized report for a wide range of breaches of normal behavior, used by the AI models trained with machine learning on the normal behavior of the system, based on analyzing previous reports with one or more models trained with machine learning on assessing and populating relevant data into the incident report corresponding to each possible cyber threat. The formatting module can generate a threat incident report in the formalized report from a multitude of a dynamic human-supplied and/or machine

created templates corresponding to different types of cyber threats, each template corresponding to different types of cyber threats that vary in format, style, and standard fields in the multitude of templates. The formatting module can populate a given template with relevant data, graphs, or other information as appropriate in various specified fields, along with a ranking of a likelihood of whether that hypothesis cyber threat is supported and its threat severity level for each of the supported cyber threat hypotheses, and then output the formatted threat incident report with the ranking of each supported cyber threat hypothesis, which is presented digitally on the user interface and/or printed as the printable report.

[0159] In the next step, the assessment module with the AI classifiers, once armed with the knowledge that malicious activity is likely occurring/is associated with a given process from the analyzer module, then cooperates with the autonomous response engine 140 to take an autonomous action such as i) deny access in or out of the device or the network and/or ii) shutdown activities involving a detected malicious agent.

[0160] The autonomous response engine 140, rather than a human taking an action, can be configured to cause one or more rapid autonomous mitigation actions to be taken to counter the cyber threat. A user interface for the response engine can program the autonomous response engine 140 i) to merely make a suggested response to take to counter the cyber threat that will be presented on a display screen and/or sent by a notice to an administrator for explicit authorization when the cyber threat is detected or ii) to autonomously take a response to counter the cyber threat without a need for a human to approve the response when the cyber threat is detected. The autonomous response engine 140 will then send a notice of the autonomous response as well as display the autonomous response taken on the display screen. Example autonomous responses may include cut off connections, shutdown devices, change the privileges of users, delete and remove malicious links in emails, slow down a transfer rate, and other autonomous actions against the devices and/or users. The autonomous response engine 140 uses one or more Artificial Intelligence models that are configured to intelligently work with other third-party defense systems in that customer's network against threats. The autonomous response engine 140 can send its own protocol

commands to devices and/or take actions on its own. In addition, the autonomous response engine 140 uses the one or more Artificial Intelligence models to orchestrate with other third-party defense systems to create a unified defense response against a detected threat within or external to that customer's network. The autonomous response engine 140 can be an autonomous self-learning response coordinator that is trained specifically to control and reconfigure the actions of traditional legacy computer defenses (e.g., firewalls, switches, proxy servers, etc.) to contain threats propagated by, or enabled by, networks and the internet. The cyber threat module can cooperate with the autonomous response engine 140 to cause one or more autonomous actions in response to be taken to counter the cyber threat, improves computing devices in the system by limiting an impact of the cyber threat from consuming unauthorized CPU cycles, memory space, and power consumption in the computing devices via responding to the cyber threat without waiting for some human intervention.

[0161] The trigger module, analyzer module, assessment module, and formatting module cooperate to improve the analysis and formalized report generation with less repetition to consume CPU cycles with greater efficiency than humans repetitively going through these steps and re-duplicating steps to filter and rank the one or more supported possible cyber threat hypotheses from the possible set of cyber threat hypotheses.

[0162] The autonomous response engine 140 is configured to use one or more Application Programming Interfaces to translate desired mitigation actions for nodes (devices, user accounts, etc.) into a specific language and syntax utilized by that device, user account, etc. from potentially multiple different vendors being protected in order to send the commands and other information to cause the desired mitigation actions to change, for example, a behavior of a detected threat of a user and/or a device acting abnormal to the normal pattern of life. The selected mitigation actions on the selected nodes minimize an impact on other parts of the system being protected (e.g., devices and users) that are i) currently active in the system being protected and ii) that are not in breach of being outside the normal behavior benchmark. The autonomous response engine 140 can have a discovery module to i) discover capabilities of each node being protected device and the other cyber security devices (e.g., firewalls) in the system

being protected and ii) discover mitigation actions they can take to counter and/or contain the detected threat to the system being protected, as well as iii) discover the communications needed to initiate those mitigation actions.

[0163] For example, the autonomous response engine 140 cooperates and coordinates with an example set of network capabilities of various network devices. The network devices may have various capabilities such as identity management including setting user permissions, network security controls, firewalls denying or granting access to various ports, encryption capabilities, centralize logging, antivirus anti-malware software quarantine and immunization, patch management, etc., and also freeze any similar, for example, network activity, etc. triggering the harmful activity on the system being protected.

[0164] Accordingly, the autonomous response engine 140 will take an autonomous mitigation action to, for example, shutdown the device or user account, block login failures, perform file modifications, block network connections, restrict the transmission of certain types of data, restrict a data transmission rate, remove or restrict user permissions, etc. The autonomous response engine 140 for an email system could initiate example mitigation actions to either remedy or neutralize the tracking link, when determined to be the suspicious covert tracking link, while not stopping every email entering the email domain with a tracking link, or hold the email communication entirely if the covert tracking link is highly suspicious, and also freeze any similar, for example, email activity triggering the harmful activity on the system being protected.

[0165] The autonomous response engine 140 has a default set of autonomous mitigation actions shown on its user interface that it knows how to perform when the different types of cyber threats are equal to or above a user configurable threshold posed by this type of cyber threat. The autonomous response engine 140 is also configurable in its user interface to allow the user to augment and change what type of automatic mitigation actions, if any, the autonomous response engine 140 may take when different types of cyber threats that are equal to or above the configurable level of threat posed by a cyber threat.

[0166] Referring to Figure 6, the cyber-attack prediction engine 105 using Artificial Intelligence-based simulations is communicatively coupled to a cyber security

appliance 100, an open source (OS) database server 790, an email system 796, one or more endpoint computing devices 791A-B, and an IT network system 792 with one or more entities, over one or more networks 791/792 in the system being protected.

[0167] The cyber-attack prediction engine 105 with Artificial Intelligence-based simulations is configured to integrate with the cyber security appliance 100 and cooperate with components within the cyber security appliance 100 installed and protecting the network from cyber threats by making use of outputs, data collected, and functionality from two or more of a data store, other modules, and one or more AI models already existing in the cyber security appliance 100.

[0168] The cyber-attack prediction engine 105 may include a cyber threat generator module to generate many different types of cyber threats with the past historical attack patterns to attack the simulated system to be generated by the simulated attack module 750 that will digitally/virtually replicate the system being protected, such as a phishing email generator configured to generate one or more automated phishing emails to pentest the email defenses and/or the network defenses provided by the cyber security appliance 100. For example, the system being protected can be an email system and then the phishing email generator may be configured to cooperate with the trained AI models to customize the automated phishing emails based on the identified data points of the organization and its entities.

[0169] The email module and IT network module may use a vulnerability tracking module to track and profile, for example, versions of software and a state of patches and/or updates compared to a latest patch and/or update of the software resident on devices in the system/network. The vulnerability tracking module can supply results of the comparison of the version of software as an actual detected vulnerability for each particular node in the system being protected, which is utilized by the node exposure score generator and the cyber-attack prediction engine 105 with Artificial Intelligence-based simulations in calculating 1) the spread of a cyber threat and 2) a prioritization of remediation actions on a particular node compared to the other network nodes with actual detected vulnerabilities. The node exposure score generator is configured to also factor in whether the particular node is exposed to direct contact by an entity generating the cyber threat (when the threat is controlled from a location external to the

system e.g., network) or the particular node is downstream of a node exposed to direct contact by the entity generating the cyber threat external to the network.

[0170] The node exposure score generator and the simulated attack module 750 in the cyber-attack prediction engine 105 cooperate to run the one or more hypothetical simulations of an actual detected cyber threat incident and/or a hypothetical cyberattack incident to calculate the node paths of least resistance in the virtualized instance/modeled instance of the system being protected. The progress through the node path(s) of least resistance through the system being protected are plotted through the various simulated instances of components of the graph of the system being protected until reaching a suspected end goal of the cyber-attack scenario, all based on historic knowledge of connectivity and behavior patterns of users and devices within the system under analysis. See for example Figures 7A and 7B. The simulated attack module 750, via a simulator and/or a virtual network clone creator, can be programmed to model and work out the key paths and devices in the system (e.g., a network, with its nets and subnets,) via initially mapping out the system being protected and querying the cyber security appliance on specific's known about the system being protected by the cyber security appliance 100. The simulated attack module 750 is configured to search and query, two or more of i) a data store, ii) modules in the detection engine, and iii) the one or more Artificial Intelligence (AI) models making up the cyber security appliance 100 protecting the actual network under analysis from cyber threats, on what, i) the data store, ii) the modules, and iii) the one or more AI models in the cyber security appliance 100, already know about the nodes of the system, under analysis to create the graph of nodes of the system being protected. Thus, the cyber-attack prediction engine 105 with Artificial Intelligence-based simulations is configured to construct the graph of the virtualized version of the system from knowledge known and stored by modules, a data store, and one or more AI models of a cyber security appliance 100 protecting an actual network under analysis. The knowledge known and stored is obtained at least from ingested traffic from the actual system under analysis. Thus, the virtualized system, and its node components/accounts connecting to the network, being tested during the simulation are up to date and accurate for the time the actual system under analysis is being tested and simulated because the cyber-attack prediction engine 105 with

Artificial Intelligence-based simulations is configured to obtain actual network data collected by two or more of 1) modules, 2) a data store, and 3) one or more AI models of a cyber security appliance protecting the actual network under analysis from cyber threats. The simulated attack module 750 will make a model incorporating the actual data of the system through the simulated versions of the nodes making up that system for running simulations on the simulator. Again, a similar approach is taken when the simulated attack module 750 uses a clone creator to spin up and create a virtual clone of the system being protected with virtual machines in the cloud.

[0171] The cyber-attack prediction engine 105 with Artificial Intelligence-based simulations is configured to simulate the compromise of a spread of the cyber threat being simulated in the simulated cyber-attack scenario, based on historical and/or similar cyber threat attack patterns, between the devices connected to the virtualized network, via a calculation on an ease of transmission of the cyber threat algorithm, from 1) an originally compromised node by the cyber threat, 2) through to other virtualized/ simulated instances of components of the virtualized network, 3) until reaching a suspected end goal of the cyber-attack scenario, including key network devices. The cyber-attack prediction engine 105 with Artificial Intelligence-based simulations also calculates how likely it would be for the cyber-attack to spread to achieve either of 1) a programmable end goal of that cyber-attack scenario set by a user, or 2) set by default an end goal scripted into the selected cyber-attack scenario.

[0172] The email module and the IT network module can include a profile manager module. The profile manager module is configured to maintain a profile tag on all of the devices connecting to the actual system/network under analysis based on their behavior and security characteristics and then supply the profile tag for the devices connecting to the virtualized instance of the system/network when the construction of the graph occurs. The profile manager module is configured to maintain a profile tag for each device before the simulation is carried out; and thus, eliminates a need to search and query for known data about each device being simulated during the simulation. This also assists in running multiple simulations of the cyberattack in parallel.

[0173] The cyber-attack prediction engine 105 with Artificial Intelligence-based simulations module is configured to construct the graph of the virtualized system, e.g. a

network with its nets and subnets, where two or more of the devices connecting to the virtualized network are assigned with different weighting resistances to malicious compromise from the cyber-attack being simulated in the simulated cyber-attack scenario based on the actual cyber-attack on the virtualized instance of the network and their node vulnerability score. In addition to a weighting resistance to the cyberattack, the calculations in the model for the simulated attack module 750 factor in the knowledge of a layout and connection pattern of each particular network device in a network, an amount of connections and/or hops to other network devices in the network, how important a particular device (a key importance) determined by the function of that network device, the user(s) associated with that network device, and the location of the device within the network. Note, multiple simulations can be conducted in parallel by the orchestration module. The simulations can occur on a periodic regular basis to pentest the cyber security of the system and/or in response to a detected ongoing cyberattack in order to get ahead of the ongoing cyberattack and predict its likely future moves. Again, the graph of the virtualize instance of the system is created with two or more of 1) known characteristics of the network itself, 2) pathway connections between devices on that network, 3) security features and credentials of devices and/or their associated users, and 4) behavioral characteristics of the devices and/or their associated users connecting to that network, which all of this information is obtained from what was already know about the network from the cyber security appliance.

[0174] During an ongoing cyberattack, the simulated attack module 750 is configured to run the one or more hypothetical simulations of the detected cyber threat incident and feed details of a detected incident by a cyber threat module in the detection engine into the collections module of the cyber-attack prediction engine 105 using Artificial Intelligence-based simulations. The simulated attack module 750 is configured to run one or more hypothetical simulations of that detected incident in order to predict and assist in the triggering an autonomous response by the autonomous response engine 140 and then restoration by the restoration engine to the detected incident.

[0175] The simulated attack module 750 ingests the information for the purposes of modeling and simulating a potential cyberattacks against the network and routes that an attacker would take through the network. The simulated attack module 750 can

construct the graph of nodes with information to i) understand an importance of network nodes in the network compared to other network nodes in the network, and ii) to determine key pathways within the network and vulnerable network nodes in the network that a cyber-attack would use during the cyber-attack, via modeling the cyber-attack on at least one of 1) a simulated device version and 2) a virtual device version of the system being protected under analysis. Correspondingly, the calculated likelihood of the compromise and timeframes for the spread of the cyberattack is tailored and accurate to each actual device/user account (e.g., node) being simulated in the system because the cyber-attack scenario is based upon security credentials and behavior characteristics from actual traffic data fed to the modules, data store, and AI models of the cyber security appliance.

[0176] The cyber-attack prediction engine 105 with its Artificial Intelligence trained on how to conduct and perform cyberattack in a simulation in either a simulator or in a clone creator spinning up virtual instances on virtual machines will take a sequence of actions and then evaluate the actual impact after each action in the sequence, in order to yield a best possible result to contain/mitigate the detected threat while minimizing the impact on other network devices and users that are i) currently active and ii) not in breach, from different possible actions to take. Again, multiple simulations can be run in parallel so that the different sequences of mitigation actions and restoration actions can be evaluated essentially simultaneously. The cyber-attack prediction engine 105 with Artificial Intelligence-based simulations in the cyber-attack prediction engine 105 is configured to use one or more mathematical functions to generate a score and/or likelihood for each of the possible actions and/or sequence of multiple possible actions that can be taken in order to determine which set of actions to choose among many possible actions to initiate. The one or more possible actions to take and their calculated scores can be stacked against each other to factor 1) a likelihood of containing the detected threat acting abnormal with each possible set of actions, 2) a severity level of the detected threat to the network, and 3) the impact of taking each possible set of actions i) on users and ii) on devices currently active in the network not acting abnormal to the normal behavior of the network, and then communicate with the cyber threat detection engine, the autonomous response engine

140, and the cyber-security restoration engine 190, respectively, to initiate the chosen set of actions to cause a best targeted change of the behavior of the detected threat acting abnormal to the normal pattern of life on the network while minimizing the impact on other network devices and users that are i) currently active and ii) not in breach of being outside the normal behavior benchmark. The cyber-attack prediction engine cooperates with the AI models modelling a normal pattern of life for entities/nodes in the system being protected.

[0177] The simulated attack module 750 is programmed itself and can cooperate with the artificial intelligence in the restoration engine to factor an intelligent prioritization of remediation actions and which nodes (e.g., devices and user accounts) in the simulated instance of the system being protected should have a priority compared to other nodes. This can also be reported out to assist in allocating human security team personnel resources that need human or human approval to restore the nodes based on results of the one or more hypothetical simulations of the detected incident.

[0178] Note, the cyberattack simulator 105, when doing attack path modelling, does not need to not calculate every theoretically possible path from the virtualized instance of the source device to the end goal of the cyber-attack scenario but rather a set of the most likely paths, each time a hop is made from one node in the virtualized network to another device in the virtualized network, in order to reduce an amount of computing cycles needed by the one or more processing units as well as an amount of memory storage needed in the one or more non-transitory storage mediums.

[0179] Figure 8 illustrates a block diagram of an embodiment of the AI-based cyber security appliance 100 with the security awareness training system 203 and other Artificial Intelligence-based engines plugging in as an appliance platform to protect a system. The probes and detectors monitor, in this example, email activity and IT network activity to feed this data to determine what is occurring in each domain individually to their respective modules configured and trained to understand that domain's information as well as correlate causal links between these activities in these domains to supply this input into the modules of the cyber security appliance 100. The network can include various computing devices such as desktop units, laptop units,

smart phones, firewalls, network switches, routers, servers, databases, Internet gateways, etc.

[0180] Referring back to Figure 4, a computer system within a building, can use the cyber security appliance 100 to detect and thereby attempt to prevent threats to computing devices within its bounds. In this exemplary embodiment of the cyber security appliance 100 with the multiple Artificial Intelligence-based engines is implemented on a computer. The computer has the electronic hardware, modules, models, and various software processes of the cyber security appliance 100; and therefore, runs threat detection for detecting threats to the first computer system. As such, the computer system includes one or more processors arranged to run the steps of the process described herein, memory storage components required to store information related to the running of the process, as well as a network interface for collecting the required information for the probes and other sensors collecting data from the network under analysis.

[0181] The cyber security appliance 100 in the computer builds and maintains a dynamic, ever-changing model of the 'normal behavior' of each user and machine within the system. The approach is based on Bayesian mathematics, and monitors all interactions, events, and communications within the system - which computer is talking to which, files that have been created, networks that are being accessed.

[0182] For example, a second computer is-based in a company's San Francisco office and operated by a marketing employee who regularly accesses the marketing network, usually communicates with machines in the company's U.K. office in second computer system 40 between 9.30 AM and midday, and is active from about 8:30 AM until 6 PM.

[0183] The same employee virtually never accesses the employee time sheets, very rarely connects to the company's Atlanta network and has no dealings in South-East Asia. The security appliance takes all the information that is available relating to this employee and establishes a 'pattern of life' for that person and the devices used by that person in that system, which is dynamically updated as more information is gathered. The model of the normal pattern of life for an entity in the network under analysis is used as a moving benchmark, allowing the cyber security appliance 100 to

spot behavior on a system that seems to fall outside of this normal pattern of life, and flags this behavior as anomalous, requiring further investigation and/or autonomous action.

[0184] The cyber security appliance 100 is built to deal with the fact that today's attackers are getting stealthier and an attacker/malicious agent may be 'hiding' in a system to ensure that they avoid raising suspicion in an end user, such as by slowing their machine down. The Artificial Intelligence model(s) in the cyber security appliance 100 builds a sophisticated 'pattern of life' – that understands what represents normality for every person, device, and network activity in the system being protected by the cyber security appliance 100.

[0185] The self-learning algorithms in the AI can, for example, understand each node's (user account, device, etc.) in an organization's normal patterns of life in about a week, and grows more bespoke with every passing minute. Conventional AI typically relies solely on identifying threats based on historical attack data and reported techniques, requiring data to be cleansed, labelled, and moved to a centralized repository. The detection engine self-learning AI can learn "on the job" from real-world data occurring in the system and constantly evolves its understanding as the system's environment changes. The Artificial Intelligence can use machine learning algorithms to analyze patterns and 'learn' what is the 'normal behavior' of the network by analyzing data on the activity on the network at the device and employee level. The unsupervised machine learning does not need humans to supervise the learning in the model but rather discovers hidden patterns or data groupings without the need for human intervention. The unsupervised machine learning discovers the patterns and related information using the unlabeled data monitored in the system itself. Unsupervised learning algorithms can include clustering, anomaly detection, neural networks, etc. Unsupervised Learning can break down features of what it is analyzing (e.g., a network node of a device or user account), which can be useful for categorization, and then identify what else has similar or overlapping feature sets matching to what it is analyzing.

[0186] The cyber security appliance 100 can use unsupervised machine learning to works things out without pre-defined labels. In the case of sorting a series of different

entities, such as animals, the system analyzes the information and works out the different classes of animals. This allows the system to handle the unexpected and embrace uncertainty when new entities and classes are examined. The modules and models of the cyber security appliance 100 do not always know what they are looking for, but can independently classify data and detect compelling patterns.

[0187] The cyber security appliance's 100 unsupervised machine learning methods do not require training data with pre-defined labels. Instead, they are able to identify key patterns and trends in the data, without the need for human input. The advantage of unsupervised learning in this system is that it allows computers to go beyond what their programmers already know and discover previously unknown relationships. The unsupervised machine learning methods can use a probabilistic approach based on a Bayesian framework. The machine learning allows the cyber security appliance 100 to integrate a huge number of weak indicators/low threat values by themselves of potentially anomalous network behavior to produce a single clear overall measure of these correlated anomalies to determine how likely a network device is to be compromised. This probabilistic mathematical approach provides an ability to understand important information, amid the noise of the network – even when it does not know what it is looking for.

[0188] The models in the cyber security appliance 100 can use a Recursive Bayesian Estimation to combine these multiple analyzes of different measures of network behavior to generate a single overall/comprehensive picture of the state of each device, the cyber security appliance 100 takes advantage of the power of Recursive Bayesian Estimation (RBE) via an implementation of the Bayes filter.

[0189] Using RBE, the cyber security appliance 100's AI models are able to constantly adapt themselves, in a computationally efficient manner, as new information becomes available to the system. The cyber security appliance 100's AI models continually recalculate threat levels in the light of new evidence, identifying changing attack behaviors where conventional signature-based methods fall down.

[0190] Training a model can be accomplished by having the model learn good values for all of the weights and the bias for labeled examples created by the system, and in this case; starting with no labels initially. A goal of the training of the model can

be to find a set of weights and biases that have low loss, on average, across all examples.

[0191] The AI classifier can receive supervised machine learning with a labeled data set to learn to perform their task as discussed herein. An anomaly detection technique that can be used is supervised anomaly detection that requires a data set that has been labeled as "normal" and "abnormal" and involves training a classifier. Another anomaly detection technique that can be used is an unsupervised anomaly detection that detects anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal, by looking for instances that seem to fit least to the remainder of the data set. The model representing normal behavior from a given normal training data set can detect anomalies by establishing the normal pattern and then test the likelihood of a test instance under analysis to be generated by the model. Anomaly detection can identify rare items, events or observations which raise suspicions by differing significantly from the majority of the data, which includes rare objects as well as things like unexpected bursts in activity.

[0192] The methods and systems shown in the Figures and discussed in the text herein can be coded to be performed, at least in part, by one or more processing components with any portions of software stored in an executable format on a computer readable medium. Thus, any portions of the method, apparatus and system implemented as software can be stored in one or more non-transitory storage devices in an executable format to be executed by one or more processors. The computer readable storage medium may be non-transitory and does not include radio or other carrier waves. The computer readable storage medium could be, for example, a physical computer readable storage medium such as semiconductor memory or solid-state memory, magnetic tape, a removable computer diskette, a random-access memory (RAM), a read-only memory (ROM), a rigid magnetic disc, and an optical disk, such as a CD-ROM, CD-R/W or DVD. The various methods described above may also be implemented by a computer program product. The computer program product may include computer code arranged to instruct a computer to perform the functions of one or more of the various methods described above. The computer program and/or the code for performing such methods may be provided to an apparatus, such as a

computer, on a computer readable medium or computer program product. For the computer program product, a transitory computer readable medium may include radio or other carrier waves.

[0193] A computing system can be, wholly or partially, part of one or more of the server or client computing devices in accordance with some embodiments. Components of the computing system can include, but are not limited to, a processing unit having one or more processing cores, a system memory, and a system bus that couples various system components including the system memory to the processing unit.

[0194] Computing devices

[0195] Figure 9 illustrates a block diagram of an embodiment of one or more computing devices that can be a part of the Artificial Intelligence-based cyber security system including the multiple Artificial Intelligence-based engines and the security awareness training system 203 discussed herein.

[0196] The computing device may include one or more processors or processing units 620 to execute instructions, one or more memories 630-632 to store information, one or more data input components 660-663 to receive data input from a user of the computing device 600, one or more modules that include the management module, a network interface communication circuit 670 to establish a communication link to communicate with other computing devices external to the computing device, one or more sensors where an output from the sensors is used for sensing a specific triggering condition and then correspondingly generating one or more preprogrammed actions, a display screen 691 to display at least some of the information stored in the one or more memories 630-632 and other components. Note, portions of this design implemented in software 644, 645, 646 are stored in the one or more memories 630-632 and are executed by the one or more processors 620. The processing unit 620 may have one or more processing cores, which couples to a system bus 621 that couples various system components including the system memory 630. The system bus 621 may be any of several types of bus structures selected from a memory bus, an interconnect fabric, a peripheral bus, and a local bus using any of a variety of bus architectures.

[0197] Computing device 602 typically includes a variety of computing machine-readable media. Machine-readable media can be any available media that can be accessed by computing device 602 and includes both volatile and nonvolatile media, and removable and non-removable media. By way of example, and not limitation, computing machine-readable media use includes storage of information, such as computer-readable instructions, data structures, other executable software, or other data. Computer-storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other tangible medium which can be used to store the desired information, and which can be accessed by the computing device 602. Transitory media such as wireless channels are not included in the machine-readable media. Machine-readable media typically embody computer readable instructions, data structures, and other executable software. In an example, a volatile memory drive 641 is illustrated for storing portions of the operating system 644, application programs 645, other executable software 646, and program data 647.

[0198] A user may enter commands and information into the computing device 602 through input devices such as a keyboard, touchscreen, or software or hardware input buttons 662, a microphone 663, a pointing device and/or scrolling input component, such as a mouse, trackball, or touch pad 661. The microphone 663 can cooperate with speech recognition software. These and other input devices are often connected to the processing unit 620 through a user input interface 660 that is coupled to the system bus 621, but can be connected by other interface and bus structures, such as a lighting port, game port, or a universal serial bus (USB). A display monitor 691 or other type of display screen device is also connected to the system bus 621 via an interface, such as a display interface 690. In addition to the monitor 691, computing devices may also include other peripheral output devices such as speakers 697, a vibration device 699, and other output devices, which may be connected through an output peripheral interface 695.

[0199] The computing device 602 can operate in a networked environment using logical connections to one or more remote computers/client devices, such as a remote

computing system 680. The remote computing system 680 can a personal computer, a mobile computing device, a server, a router, a network PC, a peer device, or other common network node, and typically includes many or all of the elements described above relative to the computing device 602. The logical connections can include a personal area network (PAN) 672 (e.g., Bluetooth®), a local area network (LAN) 671 (e.g., Wi-Fi), and a wide area network (WAN) 673 (e.g., cellular network). Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet. A browser application and/or one or more local apps may be resident on the computing device and stored in the memory.

[0200] When used in a LAN networking environment, the computing device 602 is connected to the LAN 671 through a network interface 670, which can be, for example, a Bluetooth® or Wi-Fi adapter. When used in a WAN networking environment (e.g., Internet), the computing device 602 typically includes some means for establishing communications over the WAN 673. With respect to mobile telecommunication technologies, for example, a radio interface, which can be internal or external, can be connected to the system bus 621 via the network interface 670, or other appropriate mechanism. In a networked environment, other software depicted relative to the computing device 602, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, remote application programs 685 as reside on remote computing device 680. It will be appreciated that the network connections shown are examples and other means of establishing a communications link between the computing devices that may be used. It should be noted that the present design can be carried out on a single computing device or on a distributed system in which different portions of the present design are carried out on different parts of the distributed computing system.

[0201] Note, an application described herein includes but is not limited to software applications, mobile applications, and programs routines, objects, widgets, plug-ins that are part of an operating system application. Some portions of this description are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most

effectively convey 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. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. These algorithms can be written in a number of different software programming languages such as Python, C, C++, Java, HTTP, or other similar languages. Also, an algorithm can be implemented with lines of code in software, configured logic gates in hardware, or a combination of both. In an embodiment, the logic consists of electronic circuits that follow the rules of Boolean Logic, software that contain patterns of instructions, or any combination of both. A module may be implemented in hardware electronic components, software components, and a combination of both. A software engine is a core component of a complex system consisting of hardware and software that is capable of performing its function discretely from other portions of the entire complex system but designed to interact with the other portions of the entire complex system.

[0202] Unless specifically stated otherwise as apparent from the above discussions, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers, or other such information storage, transmission or display devices.

[0203] While the foregoing design and embodiments thereof have been provided in considerable detail, it is not the intention of the applicant(s) for the design and embodiments provided herein to be limiting. Additional adaptations and/or modifications are possible, and, in broader aspects, these adaptations and/or modifications are also encompassed. Accordingly, departures may be made from the foregoing design and

embodiments without departing from the scope afforded by the following claims, which scope is only limited by the claims when appropriately construed.

Claims

1. A cyber security system, comprising:

an analyzer module configured to determine whether a file under analysis is likely malicious or not malicious,

a transformation module configured to analyze the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and iii) then to feed the representation of the file under analysis into a Large Language Model (LLM), where the LLM is trained with masked language modelling to create a semantic understanding of the file under analysis that creates a depiction of the file that retains multiple aspects of the information in and behavioral properties about the file under analysis as an embedding, in a space that allows the analyzer module to determine whether the file under analysis is likely malicious or not malicious via how closely the file under analysis as an embedding is related to a known malicious file or a known not malicious file with similar information and behavioral properties, without having to generate some kind of hash.

2. The cyber security system of claim 1, further comprising:

a data chunk analyzer that is configured to cooperate with the LLM and the transformation module, where the data chunk analyzer is 1) trained with machine learning or 2) scripted in software code to identify different parts of the file under analysis, and then break the file under analysis into its different parts when a size of the representation of that file under analysis will not fit into a content window size requirement of the LLM.

3. The cyber security system of claim 1, further comprising:

a data chunk analyzer is configured to cooperate with the LLM and the transformation module, where the data chunk analyzer is configured i) to determine a size requirement of a content window of the LLM and ii) to put portions of the

representation of that file under analysis into portions sized small enough to fit into the size requirement of the content window of the LLM in a sequence such that the LLM can retain a context that these portions sized small enough to fit into that content window size requirement all belong to a same file.

4. The cyber security system of claim 1, further comprising:

an autonomous response engine configured to cooperate with the analyzer module to perform one or more mitigation actions to mitigate a cyber threat caused by the malicious file without a need for a human to initiate the mitigation actions, where the autonomous response engine is 1) trained with machine learning, 2) scripted in software code, or 3) a combination of both to perform one or more mitigation actions.

5. The cyber security system of claim 1, further comprising:

where the LLM trained by masked language modelling to take in the representation of the file, is implemented as i) a first LLM trained to analyze compiled files under analysis and ii) a second LLM trained to analyze non-compiled files under analysis, and

where the transformation module is further configured to determine whether the file under analysis is compiled or not compiled and then to send the representation of the file under analysis to the first LLM trained to analyze compiled files or the second LLM trained to analyze non-compiled files under analysis, as appropriate.

6. The cyber security system of claim 1, wherein the LLM is configured to feed the produced embedding on the file under analysis to at least one of 1) an AI classifier that is trained to examine the embedding to determine whether the file under analysis is likely the malicious file or not likely the malicious file or 2) a relational database that is configured to store the embedding in a clustering area within the relational database.

7. The cyber security system of claim 1, wherein the transformation module is configured to upload and send the representation of the file under analysis over a secure channel in a network to a cloud platform, where the cloud platform is configured to host the LLM and an AI classifier to create the embedding and categorize the embedding as i) not a malicious file or ii) a malicious file when a new file is identified, wherein the cloud platform is configured to send the embedding down across a network to a plurality of local networks, each local network with its own cyber security appliance protecting that local network so that the analyzer module in its own cyber security appliance can determine whether the file under analysis is likely malicious or not malicious.

8. The cyber security system of claim 1, further comprising:

a detector configured to use a Bayesian statistical inference approach to dynamically improve how abnormal event scores contribute to whether the event is an indicator of a cyber threat when new data is available based upon factoring in data collected across a fleet of cyber security appliances, each cyber security appliance containing its own analyzer module.

9. The cyber security system of claim 1, further comprising:

a threat report module configured to use a second LLM trained to analyze a threat intel report and extract behavioral data in order to 1) match the extracted behavioral data to data currently present in a network under analysis to detect whether a similar potential cyber security incident is present in the network under analysis and 2) to use the extracted behavioral data as a basis for a generation of a security incident simulation performed by a prediction engine, where the prediction engine is configured to use Artificial Intelligence algorithms trained to perform a machine-learned task of Artificial Intelligence-based simulations of cyberattacks on the network under analysis.

10. A method for a cyber security system, comprising:

providing an analyzer module to determine whether a file under analysis is likely malicious or not malicious,

providing a transformation module to analyze the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and ii) then to feed the representation of the file into a Large Language Model (LLM), and

providing the LLM trained with masked language modelling to create a semantic understanding of the file under analysis that creates a depiction of the file that retains multiple aspects of the information in and behavioral properties about the file under analysis as an embedding, in a space that allows the analyzer module to determine whether the file under analysis is likely malicious or not malicious via how closely the file under analysis as an embedding is related to a known malicious file or a known not malicious file with similar information and behavioral properties, without having to generate some kind of hash.

11. The method for the cyber security system of claim 10, further comprising:

providing a data chunk analyzer to cooperate with the LLM and the transformation module, where the data chunk analyzer is 1) trained with machine learning or 2) scripted in software code to identify different parts of the file under analysis, and then break the file under analysis into its different parts when a size of the representation of that file under analysis will not fit into a content window size requirement of the LLM.

12. The method for the cyber security system of claim 10, further comprising:

providing a data chunk analyzer to cooperate with the LLM and the transformation module, where the data chunk analyzer is configured i) to determine a size requirement of a content window of the LLM and ii) to put portions of the representation of that file under analysis into portions sized small enough to fit into the size requirement of the content window of the LLM in a sequence such that the LLM can

retain a context that these portions sized small enough to fit into that content window size requirement all belong to a same file.

13. The method for the cyber security system of claim 10, further comprising:

providing an autonomous response engine to cooperate with the analyzer module to perform one or more mitigation actions to mitigate a cyber threat caused by the malicious file without a need for a human to initiate the mitigation actions, where the autonomous response engine is 1) trained with machine learning, 2) scripted in software code, or 3) a combination of both to perform one or more mitigation actions.

14. The method for the cyber security system of claim 10, further comprising:

providing the LLM trained by masked language modelling to take in the representation of the file implemented as i) a first LLM trained to analyze compiled files under analysis and ii) a second LLM trained to analyze non-compiled files under analysis, and

providing the transformation module to determine whether the file under analysis is compiled or not compiled and then to send the representation of the file under analysis to the first LLM trained to analyze compiled files or the second LLM trained to analyze non-compiled files under analysis, as appropriate.

15. The method for the cyber security system of claim 10, further comprising:

providing the LLM to feed the produced embedding on the file under analysis to at least one of 1) an AI classifier that is trained to examine the embedding to determine whether the file under analysis is likely the malicious file or not likely the malicious file or 2) a relational database that is configured to store the embedding in a clustering area within the relational database.

16. The method for the cyber security system of claim 10, further comprising:

providing the transformation module to upload and send the representation of the file under analysis over a secure channel in a network to a cloud platform,

providing the cloud platform to host the LLM and an AI classifier to create the embedding and categorize the embedding as i) not a malicious file or ii) a malicious file when a new file is identified, and

providing the cloud platform to send the embedding down across a network to a plurality of local networks, each local network with its own cyber security appliance protecting that local network so that the analyzer module in its own cyber security appliance can determine whether the file under analysis is likely malicious or not malicious.

17. The method for the cyber security system of claim 10, further comprising:

providing a detector to use a Bayesian statistical inference approach to dynamically improve how abnormal event scores contribute to whether the event is an indicator of a cyber threat when new data is available based upon factoring in data collected across a fleet of cyber security appliances, each cyber security appliance containing its own analyzer module.

18. The method for the cyber security system of claim 10, further comprising:

providing a threat report module to use a second LLM trained to analyze a threat intel report and extract behavioral data in order to 1) match the extracted behavioral data to data currently present in a network under analysis to detect whether a similar potential cyber security incident is present in the network under analysis and 2) to use the extracted behavioral data as a basis for a generation of a security incident simulation performed by a prediction engine, where the prediction engine is configured to use Artificial Intelligence algorithms trained to perform a machine-learned task of Artificial Intelligence-based simulations of cyberattacks on the network under analysis.

19. A non-transitory memory storage device to store instructions in an executable format to be executed by one or more processors, which when executed are configured to cause a computing device to perform operations as follows, comprising:

using an analyzer module to determine whether a file under analysis is likely malicious or not malicious,

using a transformation module to analyze the file under analysis in order i) to generate a representation of the file under analysis that includes a simplified summary on information in and behavioral properties about the file under analysis and ii) then to feed the representation of the file into a Large Language Model (LLM), and

using the LLM trained with masked language modelling to create a semantic understanding of the file under analysis that creates a depiction of the file that retains multiple aspects of the information in and behavioral properties about the file under analysis as an embedding, in a space that allows the analyzer module determine whether the file under analysis is likely malicious or not malicious via how closely the file under analysis as an embedding is related to a known malicious file or a known not malicious file with similar information and behavioral properties, without having to generate some kind of hash.

20. The non-transitory memory storage device of claim 19 to store additional instructions in the executable format to be executed by the one or more processors, which when executed are configured to cause the computing device to perform additional operations as follows, comprising:

using the transformation module to determine whether the file under analysis is compiled or not compiled and then to send the representation of the file under analysis to a first LLM trained to analyze compiled files or a second LLM trained to analyze non-compiled files under analysis, as appropriate, and

using the LLM to feed the produced embedding on the file under analysis to an AI classifier that is trained to examine the embedding to determine whether the file under analysis is likely the malicious file or not likely the malicious file.

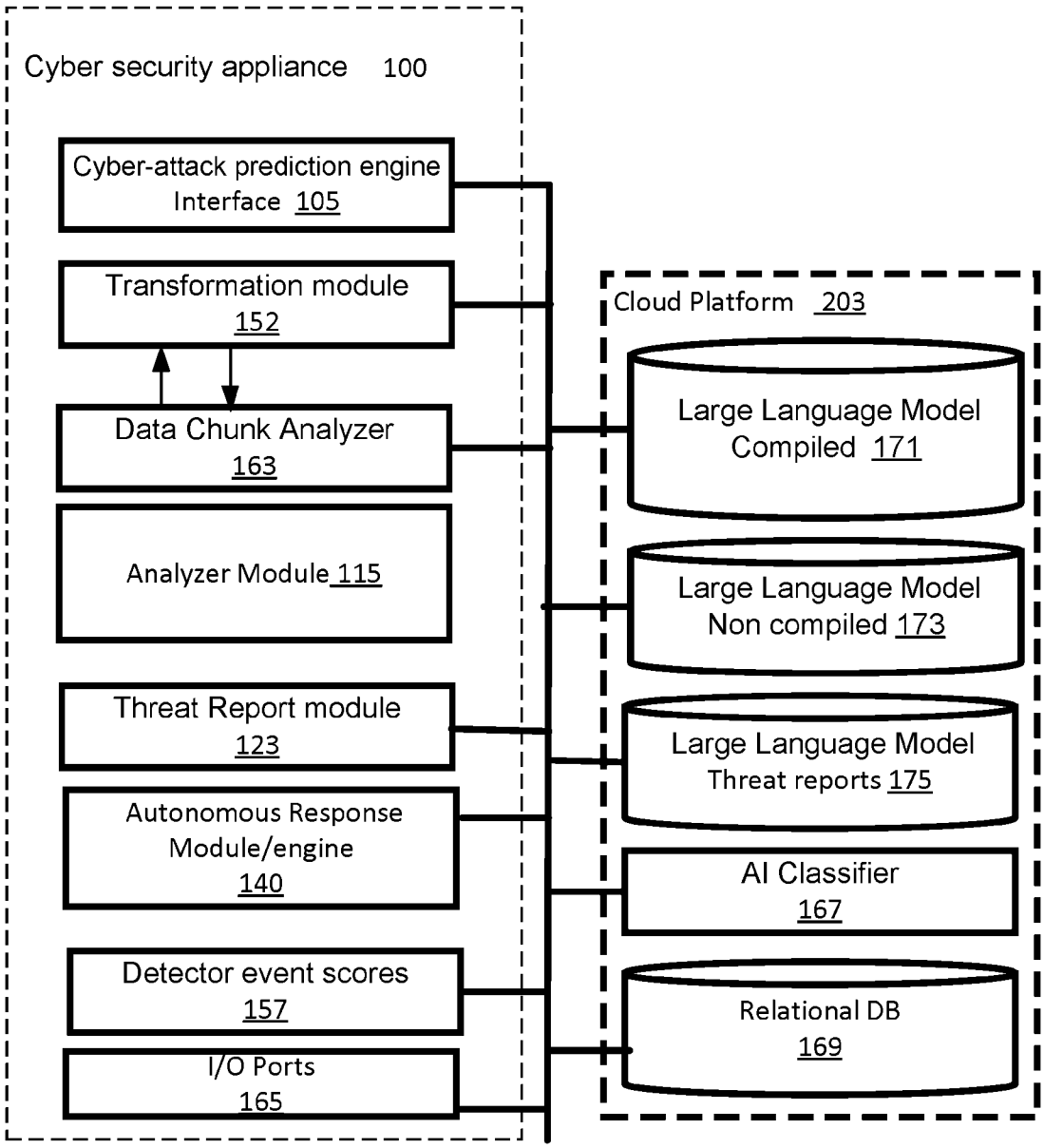
Cyber security appliance 100
Cyber-attack prediction engine Interface 105
Transformation module 152
Data Chunk Analyzer 163
Analyzer Module 115
Threat Report module 123
Autonomous Response Module/engine 140
Detector event scores 157
I/O Ports 165
Cloud Platform 203
Large Language Model Compiled 171
Large Language Model Non compiled 173
Large Language Model Threat reports 175
AI Classifier 167
Relational DB 169

Fig. 1

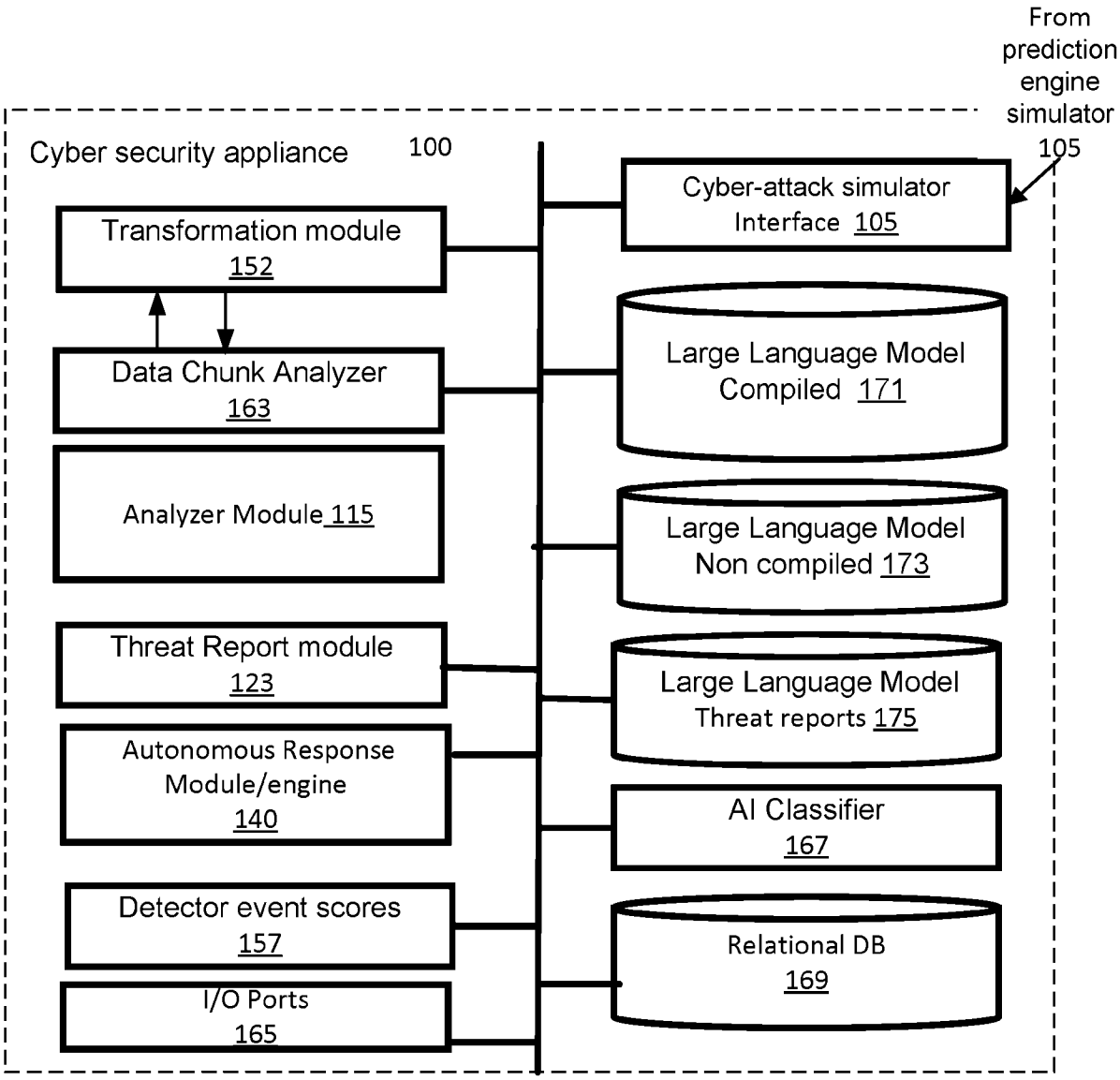
From prediction engine simulator 105
Cyber security appliance 100
Transformation module 152
Data Chunk Analyzer 163
Analyzer Module 115
Threat Report module 123
Autonomous Response Module/engine 140
Detector event scores 157
I/O Ports 165
Cyber-attack simulator Interface 105
Large Language Model Compiled 171
Large Language Model Non compiled 173
Large Language Model Threat reports 175
AI Classifier 167
Relational DB 169

Fig. 2

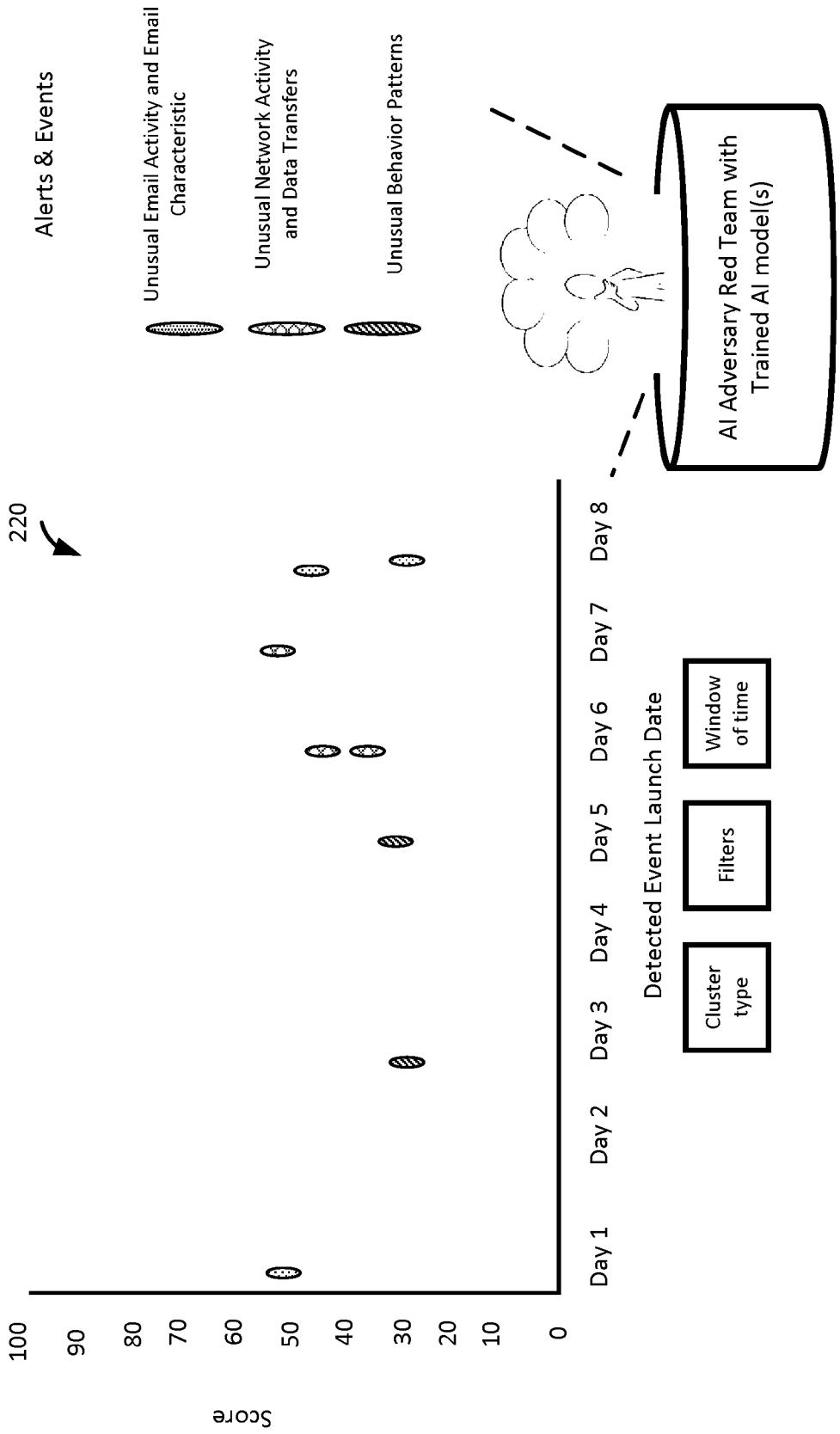
Alerts & Events
Unusual Email Activity and Email Characteristic
Unusual Network Activity and Data Transfers
Unusual Behavior Patterns
AI Adversary Red Team with Trained AI model(s)
220
Score
100
90
80
70
60
50
40
30
20
10
0
Day 1
Day 2
Day 3
Day 4
Day 5
Day 6
Day 7
Day 8
Detected Event Launch Date
Cluster type
Filters
Window of time

FIG. 3

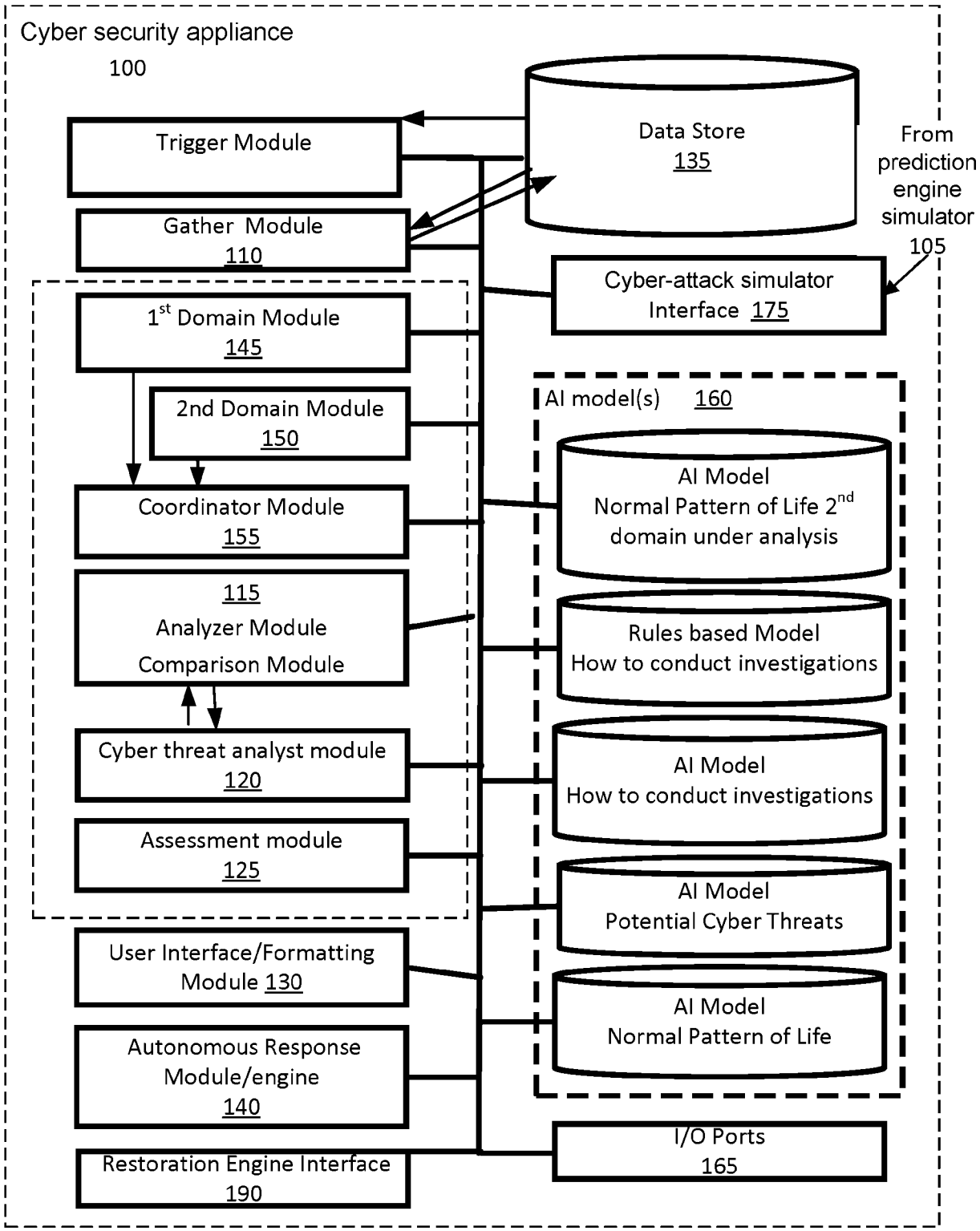
Cyber security appliance
100
Trigger Module
Data Store
135
From prediction engine simulator
105
Gather Module
110
Cyber-attack simulator Interface 175
1st Domain Module
145
2nd Domain Module
150
AI model(s) 160
AI Model
Normal Pattern of Life 2nd domain under analysis
Coordinator Module
155
115
Analyzer Module
Comparison Module
Rules based Model
How to conduct investigations
AI Model
How to conduct investigations
Cyber threat analyst module
120
Assessment module
125
AI Model
Potential Cyber Threats
User Interface/Formatting Module 130
AI Model
Normal Pattern of Life
Autonomous Response Module/engine
140
I/O Ports
165
Restoration Engine Interface
190

Fig. 4

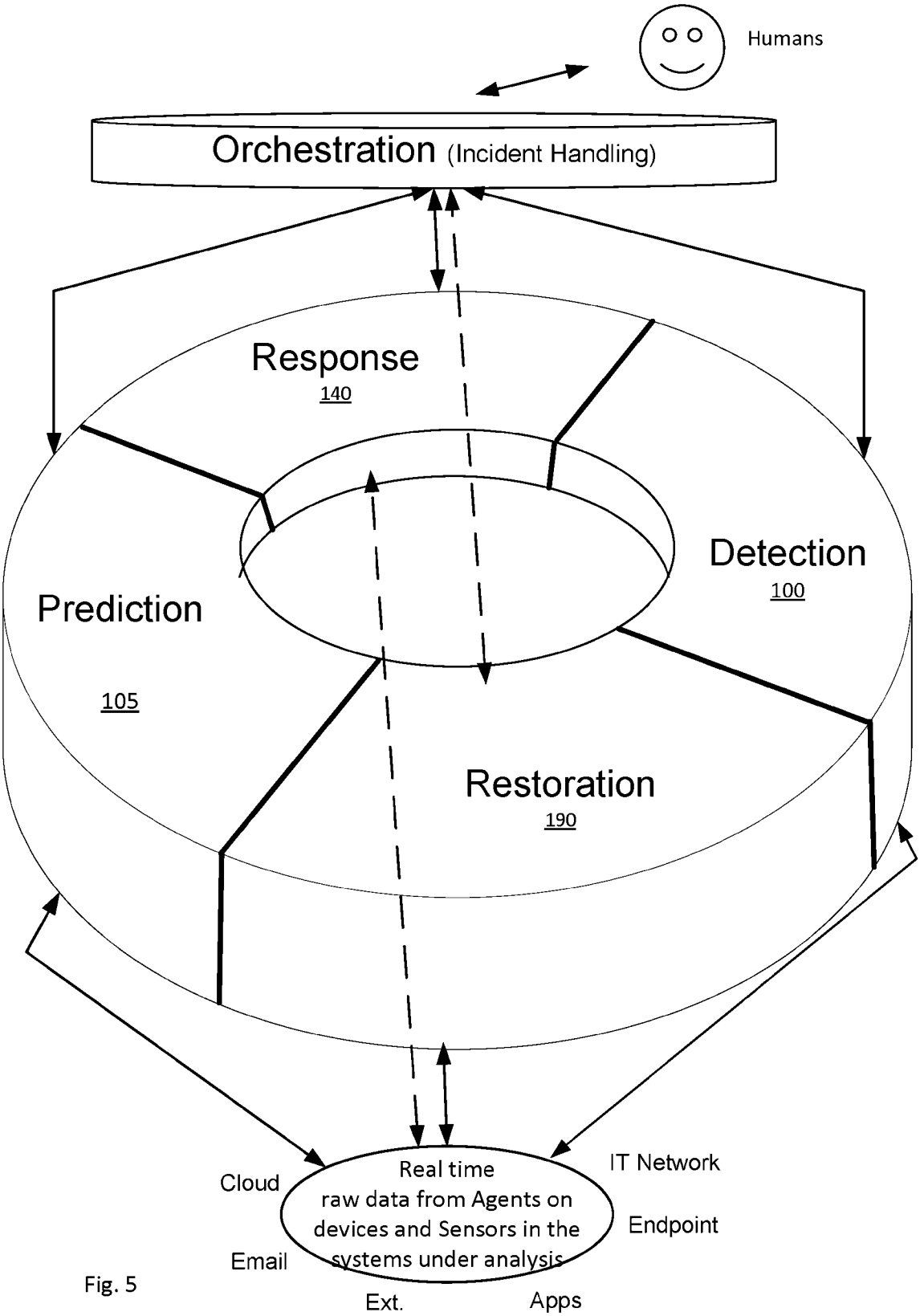
Humans
Orchestration (Incident Handling)
Response
140
Detection
100
Prediction
105
Restoration
190
Cloud
Real time
raw data from Agents on
devices and Sensors in the
systems under analysis
IT Network
Endpoint
Email
Ext.
Apps

Fig. 5

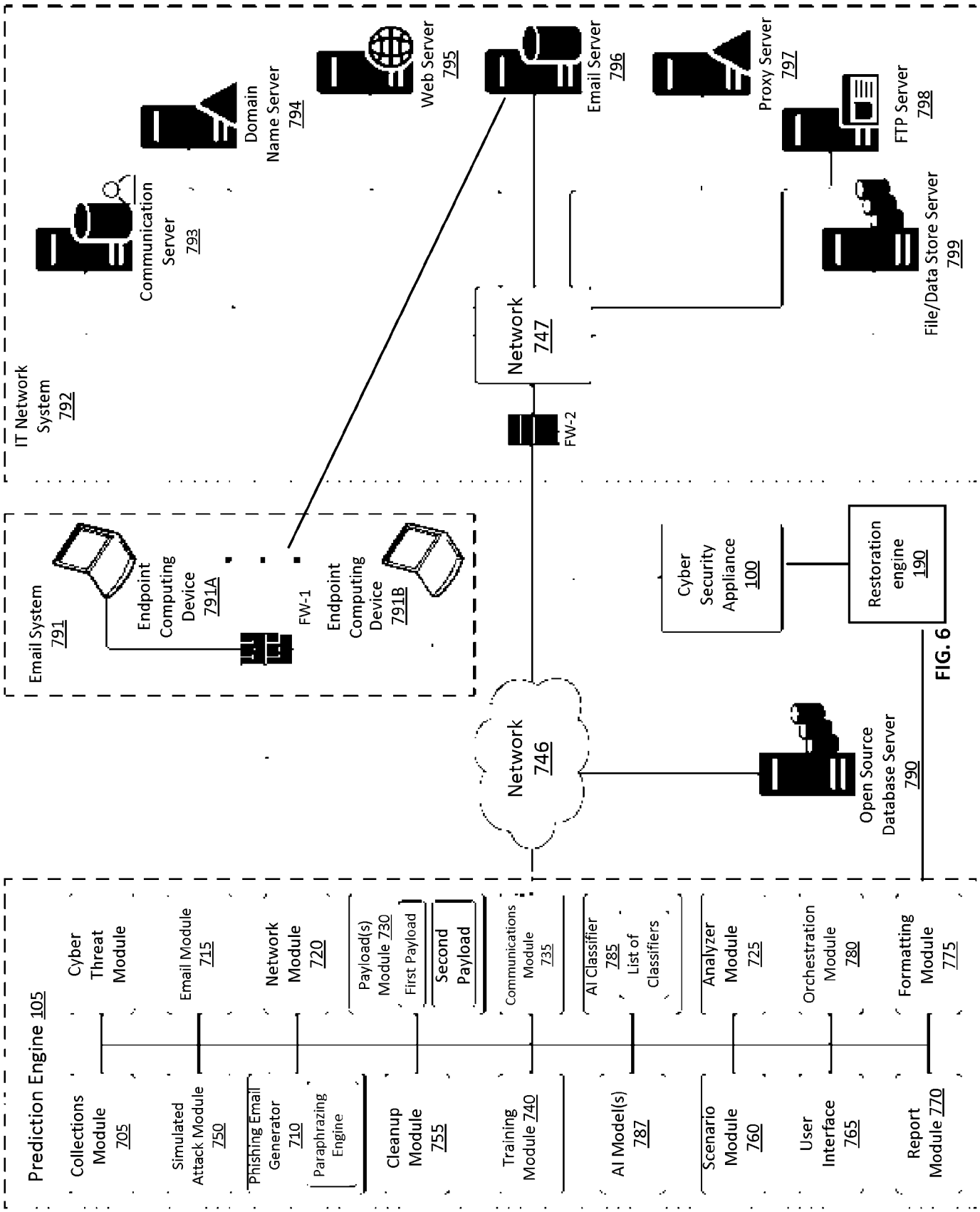
Prediction Engine 105
Collections Module 705
Cyber Threat Module
Email Module 715
Network Module 720
Payload(s) Module 730
First Payload
Second Payload
Communications Module 735
AI Classifier 785
List of Classifiers
Analyzer Module 725
Orchestration Module 780
Formatting Module 775
Simulated Attack Module 750
Phishing Email Generator 710
Paraphrazing Engine
Cleanup Module 755
Training Module 740
AI Model(s) 787
Scenario Module 760
User Interface 765
Report Module 770
Network 746
Open Source Database Server 790
Cyber Security Appliance 100
Restoration engine 190
Email System 791
Endpoint Computing Device 791A
FW-1
Endpoint Computing Device 791B
IT Network System 792
Communication Server 793
Domain Name Server 794
Web Server 795
Email Server 796
Proxy Server 797
FTP Server 798
File/Data Store Server 799
Network 747
FW-2

FIG. 6

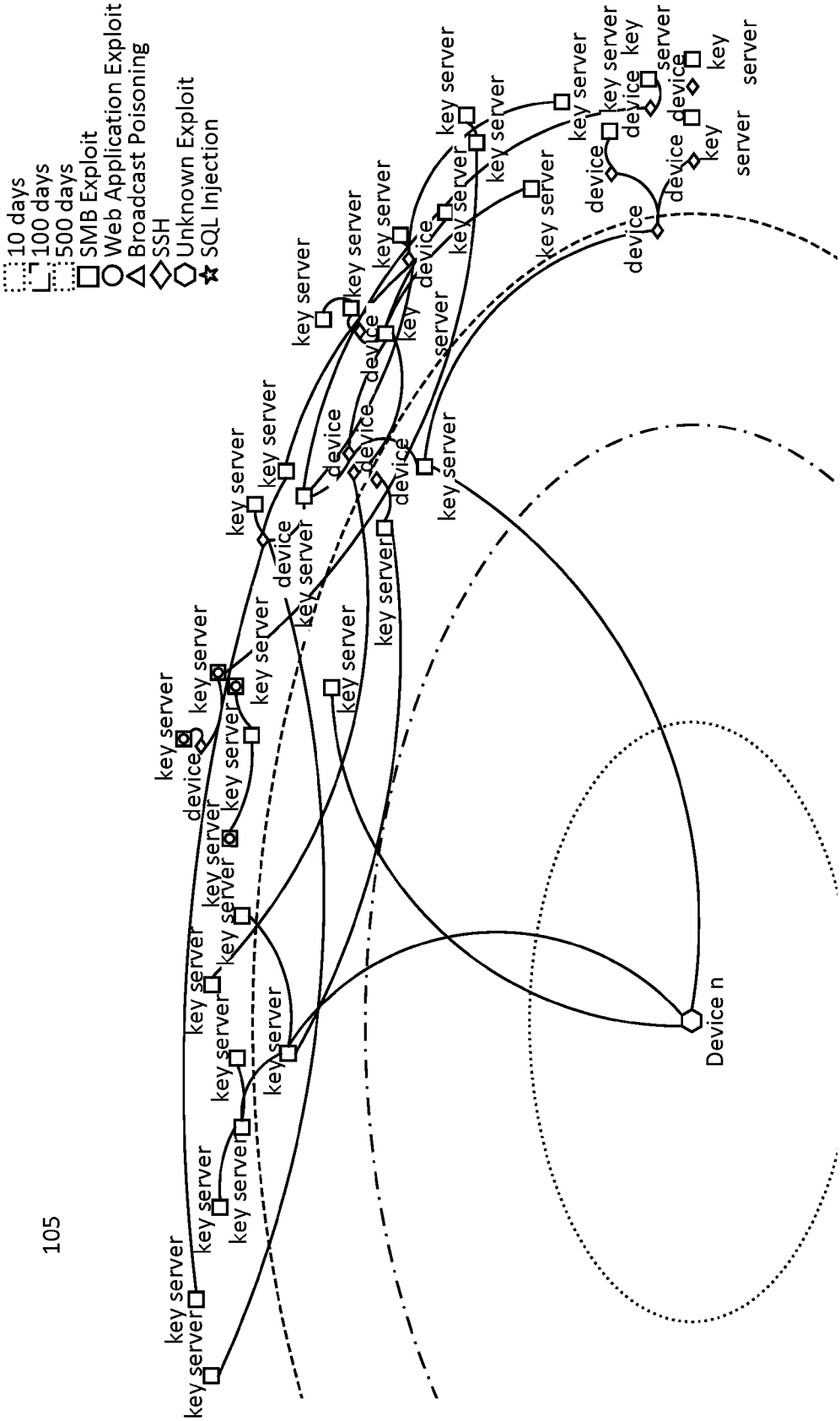
10 days
100 days
500 days
SMB Exploit
Web Application Exploit
Broadcast Poisoning
SSH
Unknown Exploit
SQL Injection
key server
device
key
server
Device n
105

Fig. 7A

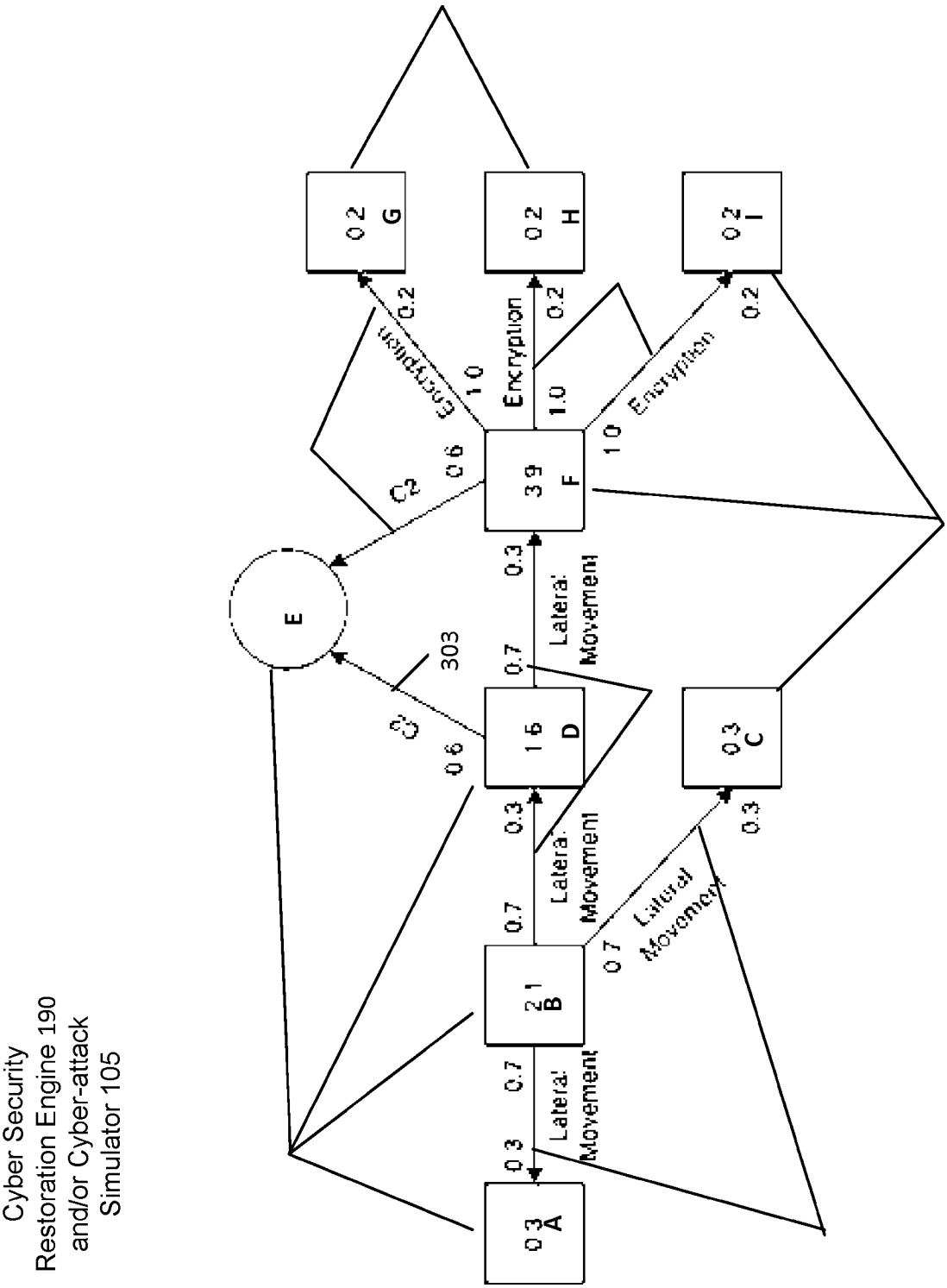
Cyber Security
Restoration Engine 190
and/or Cyber-attack
Simulator 105
0.3
A
2.1
B
1.6
D
0.3
C
3.9
F
0.2
G
0.2
H
0.2
I
E
C2
C2
303
0.6
0.6
0.7
0.3
0.7
0.3
0.7
0.3
Lateral Movement
Lateral Movement
Lateral Movement
Lateral Movement
Encryption
1.0
Encryption
1.0
Encryption
1.0
0.2
0.2
0.2

FIG 7B

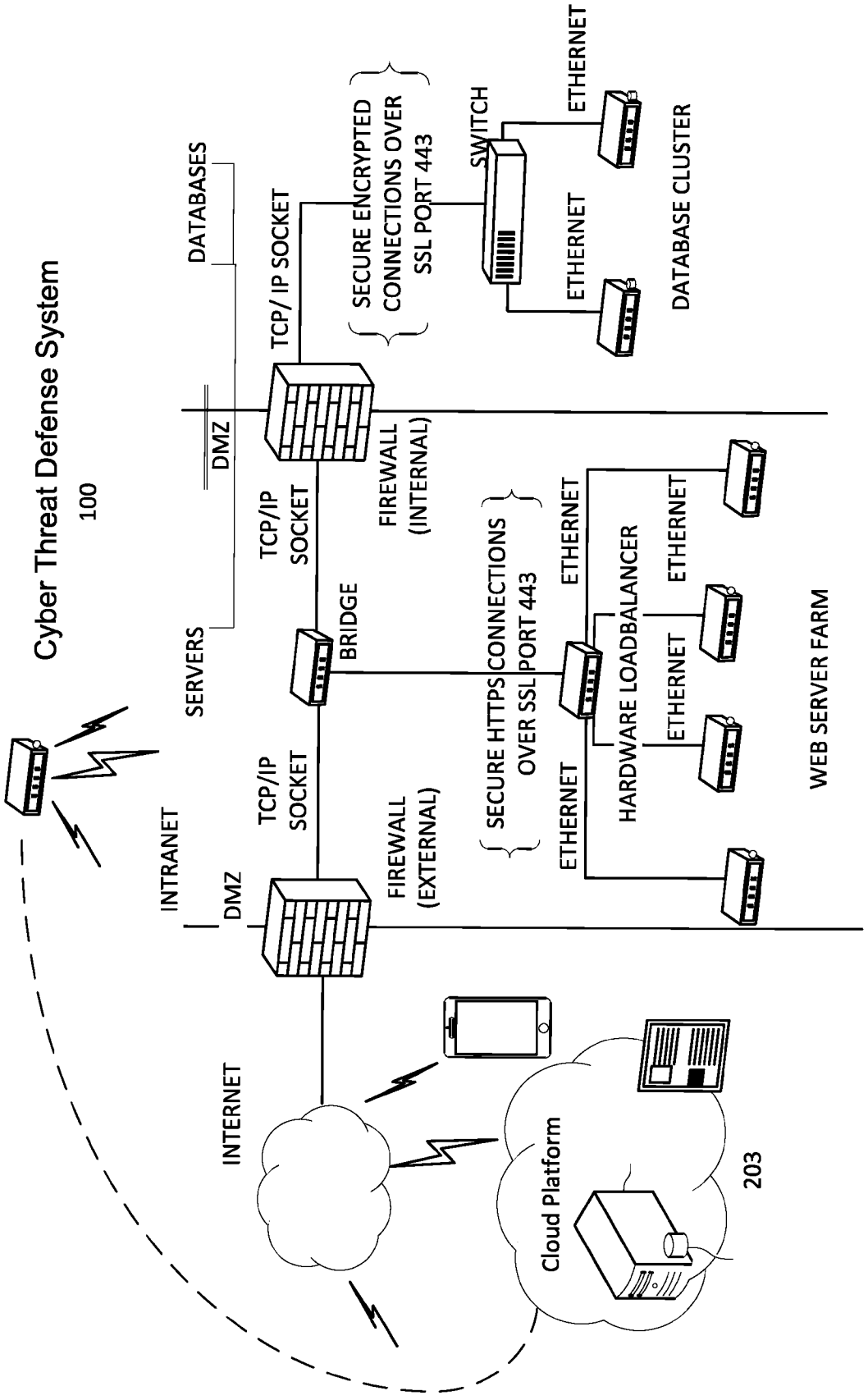

FIG. 8 Network

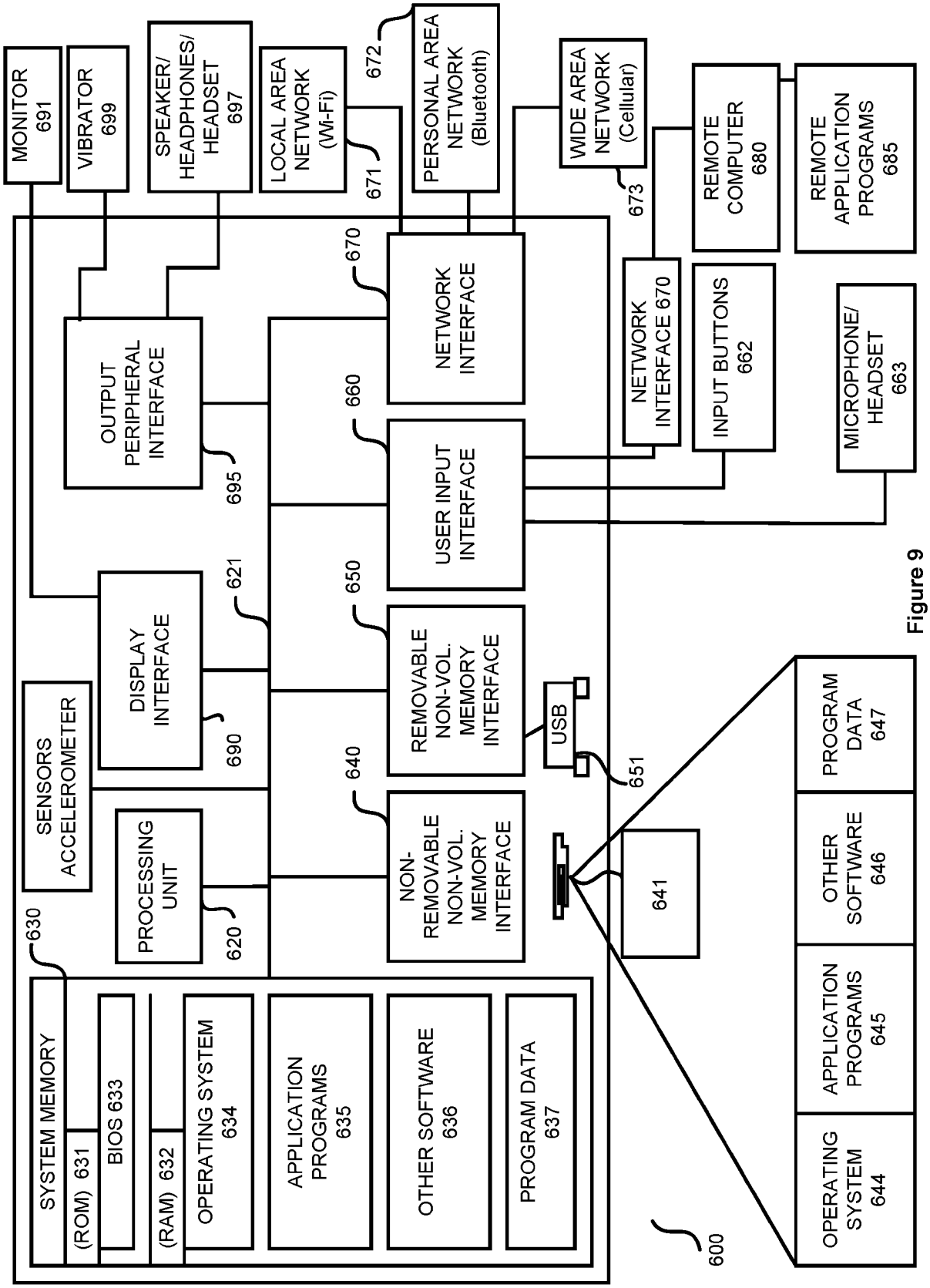
MONITOR 691
VIBRATOR 699
SPEAKER/ HEADPHONES/ HEADSET 697
LOCAL AREA NETWORK (Wi-Fi)
671
672
PERSONAL AREA NETWORK (Bluetooth)
WIDE AREA NETWORK (Cellular)
673
REMOTE COMPUTER 680
REMOTE APPLICATION PROGRAMS 685
670
NETWORK INTERFACE
NETWORK INTERFACE 670
INPUT BUTTONS 662
MICROPHONE/ HEADSET 663
OUTPUT PERIPHERAL INTERFACE
695
660
USER INPUT INTERFACE
621
650
REMOVABLE NON-VOL. MEMORY INTERFACE
USB
651
DISPLAY INTERFACE
690
SENSORS ACCELEROMETER
640
NON-REMOVABLE NON-VOL. MEMORY INTERFACE
641
PROCESSING UNIT
620
630
SYSTEM MEMORY
(ROM) 631
BIOS 633
(RAM) 632
OPERATING SYSTEM 634
APPLICATION PROGRAMS 635
OTHER SOFTWARE 636
PROGRAM DATA 637
PROGRAM DATA 647
OTHER SOFTWARE 646
APPLICATION PROGRAMS 645
OPERATING SYSTEM 644
600

Figure 9

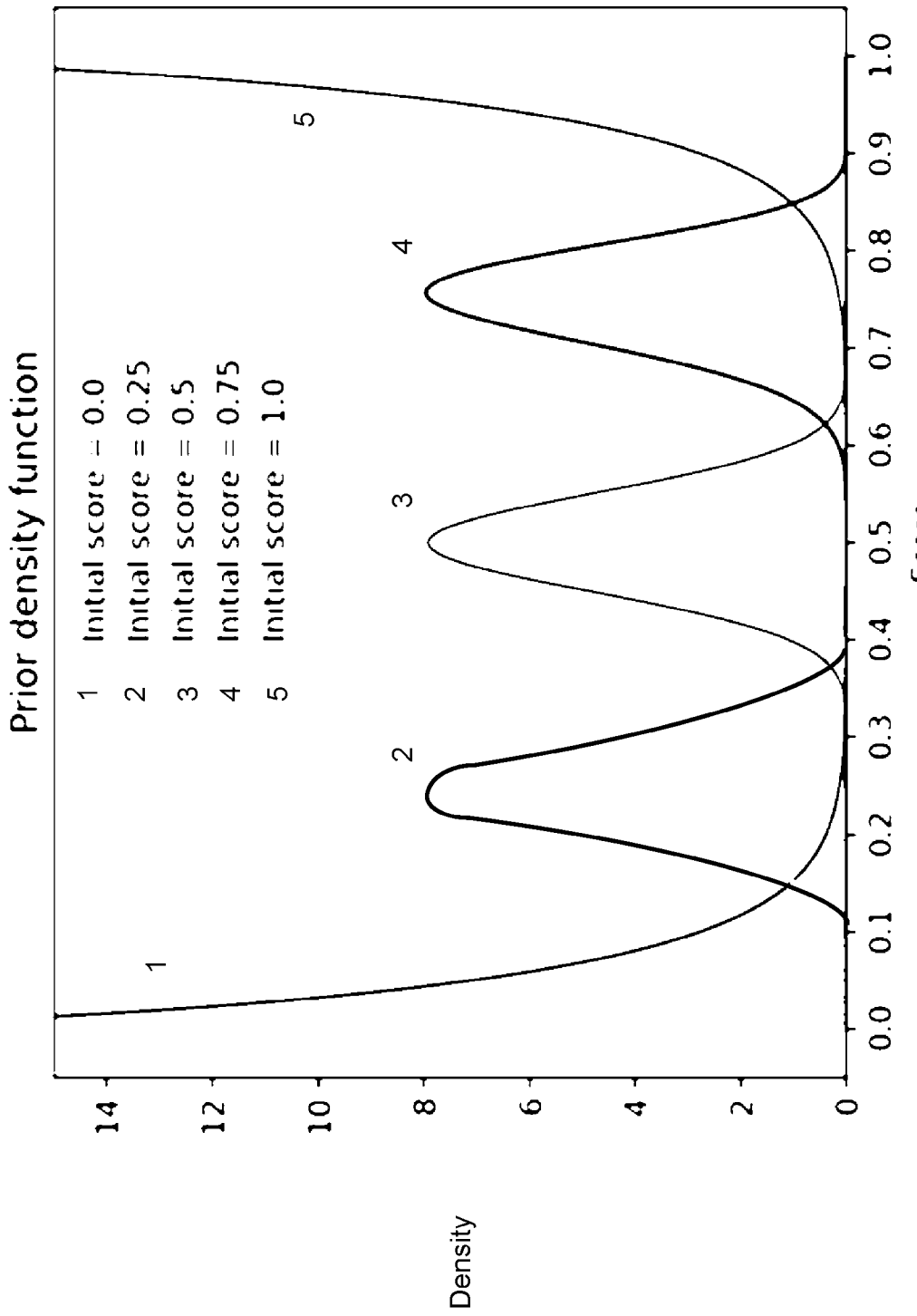
Prior density function
1 Initial score = 0.0
2 Initial score = 0.25
3 Initial score = 0.5
4 Initial score = 0.75
5 Initial score = 1.0
Density
Score
0
2
4
6
8
10
12
14
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

Figure 10

Table 1

Hostname	Counts with R=0	Counts with 1%<R<50%	Counts with 51%<R<99%	Counts with R=100%	Initial score S_0	Final Score S_f
wikipedia[.]org	12,009	756	1,404	567	0.67	0.32
tnwsdsffgc[.]shop	3	2	1	4	0.54	0.73

Figure 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2024/050192

A. CLASSIFICATION OF SUBJECT MATTER

IPC: ***H04L 9/40*** (2024.01); ***G06F 21/55*** (2024.01); G06F 21/56 (2024.01); G06N 3/045 (2024.01); G06N 3/0455 (2024.01)

CPC: ***H04L63/1483; H04L9/40; H04L63/168;*** *G06F21/554; G06F21/566; G06N3/045; G06N3/0455*

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

See Search History Document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

See Search History Document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

See Search History Document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	US 2022/0279014 A1 (MICROSOFT TECHNOLOGY LICENSING LLC) 01 September 2022 (01.09.2022)	
X	Paragraphs [0036-0037, 0045-0047, 0106].	1-2, 4, 6-7, 9-11, 13, 15-16, 18-19
Y		8, 17
A		3, 5, 12, 14, 20
	US 2022/0350883 A1 (SURIANO, V) 03 November 2022 (03.11.2022)	
Y	Paragraphs [0040-0044, 0092-0097].	8, 17
A		3, 5, 12, 14, 20
	US 2019/0312889 A1 (BANK OF AMERICA CORPORATION) 10 October 2019 (10.10.2019)	
A	Entire document.	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
03 December 2024 (03.12.2024)	**09 December 2024 (09.12.2024)**

Name and mailing address of the ISA/US	Authorized officer
COMMISSIONER FOR PATENTS **MAIL STOP PCT, ATTN: ISA/US** **P.O. Box 1450** **Alexandria, VA 22313-1450** **UNITED STATES OF AMERICA**	**SHANE THOMAS**
Facsimile No. **571-273-8300**	Telephone No. **PCT Helpdesk: (571) 272-4300**