United States Patent Application Publication	20250267154
Kind Code	A1
Publication Date	August 21, 2025
Inventor(s)	McGregor; Callum et al.

MACHINE LEARNING ANALYZING NON-STANDARD CONFIGURATIONS FOR CYBER SECURITY PURPOSES

Abstract

The DPD manager adaptively parses IT network traffic with a DPD ML engine based upon determining a port configuration setting in a network server in an IT network and a protocol utilized by IT network traffic. The DPD manager can detect a non-standard configuration set up for IT network traffic to be processed by a port on the network server, a non-standard protocol utilized by the IT network traffic, and any combination of both, and then completes a deep packet inspection upon the IT network traffic that has the non-standard configuration set up and/or the non-standard protocol utilized by the IT network traffic.

Inventors: **McGregor; Callum (Cambridge, GB), Martin; Andres Curton (Cambridge, GB), Humphrey; Dickon (Cambridge, GB)**

Applicant: **Darktrace Holdings Limited** (Cambridge, GB)

Family ID: **1000008474936**

Appl. No.: **19/058581**

Filed: **February 20, 2025**

Related U.S. Application Data

us-provisional-application US 63555823 20240220

Publication Classification

Int. Cl.: **H04L9/40** (20220101); **H04L41/16** (20220101)

U.S. Cl.:

CPC **H04L63/1416** (20130101); **H04L41/16** (20130101);

Background/Summary

NOTICE OF COPYRIGHT

[0001] A portion of this disclosure contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the material subject to copyright protection as it appears in the United States Patent & Trademark Office's patent file or records, but otherwise reserves all copyright rights whatsoever.

RELATED APPLICATION

[0002] This application claims priority under 35 USC 119 to U.S. provisional patent application No. 63/555,823, titled “CYBER SECURITY” filed Feb. 20, 2024, which is incorporated herein by reference in its entirety.

FIELD

[0003] Cyber security and in an embodiment use of Artificial Intelligence in cyber security.

BACKGROUND

[0004] 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, phishing attacks, ransomware, or 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.

SUMMARY

[0005] Methods, systems, and apparatus are disclosed for an Artificial Intelligence-based cyber security system. In an embodiment, a cyber security appliance can detect a cyber threat. The DPD manager adaptively parses IT network traffic with a DPD ML engine based upon determining a port configuration setting in a network server in an IT network and a protocol utilized by IT network traffic. The DPD manager can detect a non-standard configuration set up for IT network traffic to be processed by a port on the network server, a non-standard protocol utilized by the IT network traffic, and any combination of both, and then completes a deep packet inspection upon the IT network traffic that has the non-standard configuration set up and/or the non-standard protocol utilized by the IT network traffic.

[0006] 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.

Description

DRAWINGS

[0007] The drawings refer to some embodiments of the design provided herein in which:

[0008] FIG. **1** illustrates a block diagram of an embodiment of a Deep Packet Detection (DPD) manager that adaptively parses network traffic, such as Information Technology (IT) network traffic, with a DPD machine learning (ML) engine based upon determining i) a port configuration setting in a network server in a network and ii) a protocol utilized by the network traffic, under analysis.

[0009] FIG. **2** illustrates a graph of an embodiment of the RDA ML module identifying when the cyber threat is sending IT network traffic to an external host that is part of an interactive remote desktop type session by a shape of i) an amount of active connections, ii) an amount of external data transfer, iii) over time, and iv) whether interactive remote desktop activity would be unusual.

[0010] FIGS. **3***a* and **3***b* illustrate graphs of an embodiment of the RDA ML module in identifying when the cyber threat is using interactive remote desktop activity/session in FIG. **3***a* or not and maybe just a large data upload in FIG. **3***b*.

[0011] FIG. **4** illustrates a block diagram of an embodiment of the AI-based cyber security appliance with example components, including the DPD manager and the RDA ML module, making up a detection engine that protects a system, including but not limited to a network/domain, from cyber threats.

[0012] FIG. **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 simulator 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.

[0013] FIG. **6** illustrates a block diagram of an embodiment of the cyber-attack simulator with Artificial Intelligence-based simulations conducted in the cyber-attack simulator 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.

[0014] FIG. **7**A illustrates a diagram of an embodiment of the cyber-attack simulator 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.

[0015] FIG. **7**B illustrates a diagram of an embodiment of the cyber-attack simulator 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.

[0016] FIG. **8** illustrates a block diagram of an embodiment of the AI-based cyber security appliance and other Artificial Intelligence-based engines plugging in to protect a system.

[0017] FIG. **9** 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.

[0018] FIG. **10** 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 discussed herein.

[0019] 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

[0020] 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 indication of a cyber threat is different than a second indication of a cyber threat. 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.

[0021] FIG. **1** illustrates a block diagram of an embodiment of a Deep Packet Detection (DPD) manager that adaptively parses network traffic, such as Information Technology (IT) network traffic, with a DPD machine learning (ML) engine based upon determining i) a port configuration setting in a network server such as an IT network, OT network, Cloud network, and any combination of these, and ii) a protocol utilized by network traffic, under analysis. The DPD manager **152** can adaptively parse IT network traffic (e.g. packets) with a DPD ML engine based upon determining i) a port configuration setting in a network server and/or setting in a firewall in an IT network and ii) a protocol utilized by IT network traffic, under analysis. The DPD manager **152** is configured to be capable of detecting 1) a non-standard configuration set up for IT network traffic to be processed by a port on the network server, 2) a non-standard protocol utilized by the IT network traffic, and 3) any combination of both, and then complete a deep packet inspection upon the IT network traffic that has 1) the non-standard configuration set up for IT network traffic to be processed by the port on the network server, and/or 2) the non-standard protocol utilized by the IT network traffic.

[0022] Note, a standard configuration set up for the IT network traffic to be processed by the port on the network server can be set out in an Internet Request For Comment (RFC) or other Internet traffic organization, and a non-standard configuration set up for the IT network traffic to be processed by the port on the network server has the IT network traffic being process by a different port than called out in the Internet Request For Comment or other Internet traffic organization.

[0023] Note, a standard IT protocol utilized by the IT network traffic includes some unencrypted protocols examples such as Hypertext Transfer Protocol (HTTP) processed through port **80** (used for standard web browsing, transmits data in plain text); File Transfer Protocol (FTP) processed through ports **20** (Data) and **21** (Control)—(used for transferring files between systems, unencrypted data transfer); Dynamic Host Configuration Protocol (DHCP) processed through port **67** (used for issuing IP addresses); and Network Time Protocol (NTP) processed through port **123** (used to synchronize the clocks of computers and devices across a network), Voice over Internet Protocol (VOIP) (port **56**): processed through port **56** (used for phone conversation), Remote Desktop Protocol (RDP) processed through port **3389** (used for remotely access and control another computer) as well as includes some encrypted protocols examples such as HyperText Transfer Protocol Secure (HTTPS) processed through port **443** (secure version of HTTP, uses TLS encryption for secure web browsing); Secure Shell (SSH) processed through port **22** (used for secure remote administration of devices), Secure File Transfer Protocol (SFTP) also processed through port **22** (Secure version of FTP, utilizes SSH encryption for file transfers; File Transfer Protocol over SSL (FTPS) processed through port **990** (another secure version of FTP, using TLS encryption); etc. Generally, each different protocol is associated with a specific port number on a network server to allow a system to identify the intended service of the network traffic.

[0024] In an IT network, the machine learning in the DPD manager **152** in the cyber security appliance **100** adaptively guides decision making on how to parse traffic (e.g. packets), which will improve performance for networks that contain traffic/services with non-standard traffic configurations set up on the servers and/or use of non-standard IT protocols. The Deep Packet Detection manager allows the detection of both IT network traffic in standard configurations and non-standard traffic configurations set up on the network servers, and/or use of non-standard IT protocols. The DPD manager **152** performs Deep Packet Inspection (DPI) of the IT network traffic to see whether the network servers and firewall are configured in a non-standard configuration as well as whether the IT network traffic is using a protocol that that the programmers of the DPD manager **152** are not aware of. A lot of DPI is about assumptions so hard coded values on ports, hard coded signatures, everything is statically set out in RFCs, and the DPD manager **152** is elevating network traffic analysis to be more dynamic. The DPD manager **152** does not have to make assumptions about i) the port configuration setting in the network server and/or setting in a firewall in the IT network or ii) the protocol utilized by the IT network traffic, under analysis, but rather have the DPD ML engine learn after being deployed with unsupervised machine learning but over time about which particular ports are servicing network traffic in this IT network and what protocols are being utilized by the IT network traffic in this IT network. Thus, at the start of a deployment with the DPD manager **152**, there may be some issues using unsupervised learning but over time, the DPD manager **152** learns more and more about this network's firewall and servers' configurations and protocols used with its IT network. For example, the DPD manager **152** learns and stabilizes on this network uses, for example, a HTTP server. Therefore, all the network traffic going to this network has a high probably of using the HTTP and/or HTTPS protocols and determines this in real time with DPI; rather than, maybe after hours of being in monitoring mode, the DPD manager **152** will do more than merely analyzing traffic.

[0025] The DPD manager **152** has a packet inspection hierarchical structure. The DPD manager **152** can use a set of DPD Branches. Each DPD Branch has a DPD Subbranch that has its own protocol analyzer state component, port state component, signature state library, and DPD ML engine. Each protocol analyzer state component can factor in the analyzer context, analyzer tags, and analyzer configurations. Each port state component can factor in port context, port number and function, and port configuration. The port state component associates the ports and their protocols used in this particular network. Each signature state library can factor in a signature tag, signature conditions, actions, and a pattern database. The signature state library stores a bunch of signatures that can be used to identify the existing protocols. Each DPD ML engine can factor in port predictions with individual port predictions, protocol characteristics, and analyzer tag likelihood. The DPD ML engine can cooperate with the protocol analyzer state component, port state component, and signature state library to make their respective decisions.

[0026] The DPD manager **152** is configured to understand that there are a lot of suggested RFC and other standards on the Internet about how an IT team should set up servers to process network traffic. For example, a server configured for unencrypted traffic in HTTP should be on Port **80**, and encrypted traffic should be on Port **443**. However, the DPD manager **152** is configured to understand that there is no actual requirement in any servers for an IT team to always set up servers to process network traffic in accordance with a particular RFC. The IT team can just route different types of IT network traffic on any port they choose to designate. Each network configuration when set up by its IT team can put different types of IT network traffic on any port especially if like in the cloud, or if that port is already taken, etc. The IT team might have to customize this network and just pick a random port to try and put their IT network traffic through. The DPD manager **152** can have a protocol analyzer state component to check a setup of whether this network has a standard port configuration set up on their servers for passing different types of IT network traffic as published by a suggested RFC or other standard about network traffic or does this network have some customized non-standard port configuration for passing IT traffic.

[0027] Versions of protocols (e.g. HTTP version 1.0, HTTP version 1.1, HTTP version 2, and HTTP version 3)

[0028] The DPD manager **152** can use its protocol analyzer state component, port state component, signature state library, and DPD ML engine to check whether the system recognizes the protocol or at least partially recognizes the protocol. The first check is ‘Does the Protocol analyzer state component fully recognize the protocol, or does it not fully recognize the protocol?’ If No, the secondary check between the protocol analyzer state component, the port state component, the signature state library, and the DPD ML engine determines if we do not fully recognize the protocol, then does the DPD ML engine think it's something similar to a previous protocol that that we've seen or stored in the signature state library, or not; and thus, partially recognize the protocol (e.g. HTTP version 1.0 compared to HTTP version 3.0). The DPD ML engine determines ‘Does the system partially recognize enough of the data under analysis from the protocol so that the system can deduce that this is a variant of a known protocol in our library of protocols (e.g. HTTP version 1.0 compared to HTTP version 3.0 and/or HTTP version 1.0 compared to HTTPS using TLS 1.2). The DPD manager **152** and its components analyze these packets using the library of protocols in the protocol analyzer state component and the library of meta data associated with protocols stored in the signature state library.

[0029] The DPD manager **152** can use a set of DPD Branches. Each DPD Branch has a DPD Subbranch that has its own protocol analyzer state component, port state component, signature state library, and DPD ML engine. Each protocol analyzer state component can factor in analyzer context, analyzer tags, and analyzer configurations. Each port state component can factor in port context, port number and function, and port configuration. Each signature state library can factor in a signature tag, signature conditions, actions, and a pattern database. Each DPD ML engine can factor in port predictions with individual port predictions, protocol characteristics, and analyzer tag likelihood.

[0030] The DPD manager **152** uses machine learning. The DPD manager **152** uses a set of DPD Branches. Each DPD Branch has a DPD Subbranch that has its own protocol analyzer state component, port state component, signature state library, and DPD ML engine. The DPD manager **152** has DPD subbranches for the different environments that are being analyzed. For example, encrypted traffic and unencrypted traffic can be analyzed separately. In some examples, DPD subbranches can be divided into, for example, TCP traffic, UDP traffic, etc., and then each subbranch does traffic header and file/content analysis on that type of traffic.

[0031] Different protocols are text based publications for the most part; and thus, the DPD ML engine can use large language model training to understand the particulars and generalities of each different type of protocol (e.g. HTTP compared to RDP) Thus, the protocol analyzer state component analyzes the IT network traffic and the protocol it is using but sometimes the produced results will have, e.g. English words that sometimes it doesn't yet understand or make sense in context. Thus, if the protocol analyzer state component analyzes the IT network traffic and the protocol and understands, e.g. 80% of the traffic is using words and traffic that we do fully understand, then the protocol analyzer state component can pass the IT network traffic in its protocol being utilized and the protocol analyzer state component's analysis to the DPD ML engine. The DPD ML engine can infer that this is maybe just a different version because it looks so similar enough. And then, once the DPD manager **152** sends it over to the intelligence of the DPD ML engine to deduce what that protocol is, then the DPD manager **152** will then store that deduced protocol for that IT traffic to be analyzed on whatever port by the port state component, that will store this IT traffic and associated protocol and port for a subsequent packet being analyzed. The port state component updates the analyzed ports and protocols. The signature state library updates the signatures tag, conditions, actions, and the pattern database. The DPD manager **152** is further configured to create a [pseudo] matrix in a memory of the cyber security appliance **100** of all of the ports servicing network traffic in this IT network and then analyze for and store in the memory metadata associated with network traffic being processed by each port, which is then fed to the DPD ML engine to deduce and predict what type of IT network traffic is being process by each port for [this particular] the IT network.

[0032] The DPD manager **152** uses at least a signature state library, protocol analyzer state component, and adaptive intelligence DPD ML engine component system and other components to perform a deep packet inspection of the IT network traffic. The adaptive DPD ML engine can also be trained to adaptively guide decisions on how to parse packets when the DPD manager **152** encounters some IT traffic with non-standard protocol configurations. The DPD ML engine tries to figure out what is going on with the IT traffic with non-standard protocol configurations, and then stores it best guest results into the signature state library and its pattern database, in case the DPD manager **152** runs into that same situation again.

[0033] The protocol analyzer state component, the port state component, the signature state library, and the DPD ML engine cooperate to perform a deep packet inspection of network traffic by inspecting the packet header information, the content of data packet, etc., and turns this packet analysis into metadata, which then can be fed and/or fetched to the DPD ML engine. The deep packet inspection is relied upon to detect what the type of traffic is under analysis. Most traffic types obey ports and protocols dictated by published standards, for example, RFCs. For example, HTTP is suggested to always be processed on Port **80** or **43**. However, some IT teams will utilize network traffic that has a non-standard configuration. The DPD manager **152** can be configured to I) cooperate and receive input from network sensors configured to perform deep packet inspection upon the IT network traffic in order to detect and determine 1) a standard configuration set up for IT network traffic to be processed by the port on the network server and the standard IT protocol utilized by the IT network traffic as well as II) cooperate and feed in meta data to the DPD ML engine to determine 1) the non-standard configuration set up for IT network traffic to be processed by the port on the network server, 2) the non-standard IT protocol utilized by the IT network traffic, and 3) any combination of both. If the parsing by the network sensors configured to perform deep packet inspection upon the IT network traffic is unsuccessful, the unusual traffic is then processed by the DPD manager **152** which will check that packet of network traffic against all of the library list of protocols. The DPD manager **152** can use a protocol analyzer state component to check the non-standard IT protocol utilized by the IT network traffic against all of the known protocols for the IT network traffic in a library of protocols used by the protocol analyzer state component. Each protocol analyzer state component can use a trained AI classifier to check the IT network traffic against these known protocols. When a match is found that successfully turns the packet, under analysis, into valid metadata and a coherent decoding of that packet as determined by the DPD ML engine, then no further analysis is needed but merely storing those results in the port state component and signature state library. When no match occurs, then the protocol analyzer state component feeds the metadata and at least the partially recognized characteristics of a given known protocol over to the DPD ML engine to deduce and predict what type of protocol is being used by the IT network traffic, under analysis. The unusual traffic is analysed against every single protocol until one of them matches or produces a valid output. And then, the signature state library will store that relationship of the unusual traffic under analysis to the protocol to the port relationship. Next time, when a new unusual packet is seen on that port, the pipeline of the DPD manager **152** can rapidly pull up the newly made mapping record(s) prior to going through the entire protocol matching checking process. Note, the DPD manager **152** may not be able to initially recognize the unusual traffic under analysis. However, the DPD ML engine can analyse some meta data such as a probability that the best analyser starts analysing with is HTTP because this packet had a lot of similarities to permutations of HTTP, which were considered standard. Thus, the DPD manager **152** performing the deep packet inspection may suspect that the DPI is not working in this case because it is actually a different permutation of the same protocol. For example, this traffic might be using HTTP version 2 verses HTTP version 1. The DPD manager **152** with its protocol analyzer state component can supply as meta data its best guess of what analyser to start with. The input from network sensors performing a deep packet inspection in the network and the DPD manager **152** cooperate to detect protocols running on standard ports as well as non-standard ports to pass the traffic faster. The DPD manager **152** ensures that there is no IT network traffic missed/not analyzed when unusual traffic is attempting to pass in the network being protected by the cyber security appliance **100**. The DPD manager **152** and its components allow the system to encapsulate and expand a capacity to deal with more in depth and varied traffic analysis that most systems normally do not have the resources to deal. The way DPI works in other systems they can sometimes cause a loss of visibility into traffic when it is delivered on other than dedicated ports suggested by a RFC, which can cause a false positives alert on that traffic. However, the DPD manager **152** allows for IT teams to customize their network configurations (with ports associated with a particular traffic type and/or traffic using a particular protocol that is unusual) and reduce false positive alerts on that traffic.

[0034] The protocol analyzer state component, the port state component, and signature state library can cooperate to use a large library of DPI protocol parsers and analysers. When a packet is received, the protocol analyzer state component, the port state component, and the signature state library will try to analyse this packet in the network traffic using the corresponding parser for that known port. If the parsing is successful and valid metadata is returned, the DPD ML engine need not be called. The network traffic using a standard protocol for the port configuration for that server is processed. If the parsing is unsuccessful, the packet in the network traffic is processed by the protocol analyzer state component, the port state component, the signature state library, and the DPD ML engine go through the library of all known protocol analysers and apply them until a match is found to successfully turn the packet into valid metadata. When a match is found, the port state component stores this protocol-port relationship for future use by the main parsing pipeline. The DPD manager **152** can then make a mapping of the particular ports that are servicing network traffic in this IT network and what protocols are being utilized by the IT network traffic in this IT network. The DPD manager **152** can use the port state component to store a relationship of the non-standard IT protocol utilized by the IT network traffic, under analysis, to a deduced protocol being used by the IT network traffic by the DPD Machine ML engine and an associated port being used by the IT network traffic using the non-standard IT protocol, and when a subsequent packet in network traffic is seen on this port, then the DPD manager **152** can subsequently rely upon the port state component to provide the deduced protocol being used by the IT network traffic and the associated port being used by the IT network traffic. When in the future this mapping does not actually successfully parse the traffic, this failure is then sent to the DPD ML engine and the port state component so that the mapping by the port state component will be de-prioritized in future and eventually reset to a new protocol being used. The DPD manager **152** will then repeat the earlier steps. Eventually, the DPD manager **152** builds up a library of mappings, and when taken across all of the network environments protected by the cyber security appliance **100**, the highest probability protocol mappings for each port. Note, a protocol analyzer state can describe what a protocol should be, and then a signature state library describes the meta data of what the protocol physically is. Additionally, the DPD manager **152** can use the port state component to keep track of and analyze what ports are being looked at and use the DPD machine learning engine to do port predictions.

[0035] The DPD manager **152** of the detect engine analyzes network traffic with a deep packet inspection not just on a connection level but rather also on contents within a packet level. The DPD manager **152** uses adaptive AI machine learning on how to parse that traffic so that it can be now particularized or customized to that particular implementation customer implementation to whatever this network's configuration and typical protocol setup is. The DPD manager **152** uses adaptive AI machine learning on a per customer's network deployment to start off with no assumptions. The DPD manager **152** builds up the knowledge base for future assumptions based on what actual network traffic, detection of protocols utilized, and server and firewall configurations we've actually seen and observed in this particular network rather than what some designer thinks the DPD manager **152** is going to see when deployed. The DPD manager **152** when deployed, after, for example, couple week learning period, the DPD manager **152** can start signalling 'normal network traffic' as unusual when what might be 'normal network traffic' for most networks on the internet, it is specifically unusual for this customer network environment. The DPD manager **152** learns about this network environment to stabilize on known truths about this network and then use that knowledge to impact future analysis. Thus, learns per deployment to include information that is specific to this customer and over time excludes all of the knowledge about the world that is not relevant to this customer network deployment and gets even more specific to this customer, so it kind of prevents false positives and also false negatives. However, prior network analysis components must make assumptions about networks, either guided by what is in Internet RFCs or universal networking behaviours, but in many existing cases, these are wrong. The DPD manager **152** can cooperate with the network analysis components to analyze traffic with these Internet RFCs or universal networking behaviours. However, when the network traffic analysis determines that Internet RFCs or universal networking behaviours are not being used then the DPD manager **152** uses machine learning to guide the decision making on how to parse traffic, this will improve performance for customer deployments that contain traffic/services with non-standard configurations. This could be e.g., having lots of HTTP traffic on port **1234** instead of the normal port **80**/**443**. The DPD manager **152** allows the detect engine to analyze the network traffic in a real time environment with the dynamic detection of protocols on non-standard protocols, possibly with non-standard configurations over time.

[0036] Note instructions implemented in software for the DPD manager **152**, the DPD ML engine, and its other components can be configured to be stored in one or more non-transitory storage mediums to be executed by one or more processing units.

[0037] The DPD manager **152** can be implemented via basic machine learning with some customizations discussed below and can take any input from the rest of the modules and models in the cyber security appliance **100**.

[0038] In the DPD manager **152**, the training of the machine learning for the DPI is basic on how to perform the different tasks of deep packet inspection and protocol analysis but learns the particulars of what is actually occurring within that network with on the job training. The training of the machine learning can learn these are normal standard configurations, and what the traffic normally looks like but does not make an assumption that the servers and firewall are configured in a standard way or that the network traffic is using a standard protocol. The DPD ML engine can learn through an API communication telling it 'No,' this is wrong, or 'Yes' this is correct as well as a specific reason why you are wrong, or you are correct. The DPD ML engine can build up its internal database, and then the API allows a user to delete the database at any point and restart it. The DPD ML engine can undergo a continuous, ever changing training phase and a learning phase using unsupervised learning and Bayesian probabilities and having a database. As time goes on-protocols get newer versions or the system might see a recently created protocol, and then the DPD ML engine can start unlearning the old one and learning the new one.

[0039] The DPD manager **152** can provide one more feature. The DPD manager **152** need not see an entire connection to perform its analysis. So normally, most connections start off with a handshake between a client and a server for them to exchange that information, and quite often we can lose that handshake. And, most of the time you then just have to drop the connection because you don't know whether you're reading random bytes of a file, or whether you're actually reading the metadata. The DPD manager **152** can do this when the system can assume a protocol, the system can kind of have some level of midstream analysis of network traffic after the handshake has already occurred. The DPD manager **152** can jump in and pick apart the packet to work out, if we can salvage it in any way. The DPD manager **152** can do this because it has a system to make a good guess of what the protocol is for the network traffic under analysis; otherwise, you might get some garbage data out.

[0040] A remote desktop activity (RDA) machine learning module **154** can work with and supplement the DPD manager **152** to assist in identifying when the cyber threat, such as an advanced persistent threat, is sending/uploading IT network traffic to an external host that is part of interactive remote desktop activity. The RDA machine learning module can look at a shape of i) an amount of active connections, such as two active connections, three active connections, etc., ii) an amount of data transfer, such as ten megabytes, fifteen megabytes, etc., iii) over time such as hourly increments 10 PM, 11 PM, etc., and iv) whether an RDP session would be unusual for the IT network and its users, under analysis.

[0041] The RDA ML module **154** can detect remote management activity not only in the RDP protocol but in any other protocols where an active remote management session/interactive remote desktop activity is being performed. For example, an active remote management using AnyDesk, TeamViewer, etc. that uses the protocol of HTTPS. The key advantage of the RDA ML module **154** is that it is able to distinguish these interactive remote desktop sessions from other types of activity, for example data transfers to backup services, video streaming, etc. The RDA ML module **154** can identify when an interactive remote (desktop) session is opened over encrypted protocols like HTTPS using something like TeamViewer, and then additionally tries to establish whether that interactive remote desktop session is “legitimate” as these tools are often used in IT support scenarios (as well as cyber incidents).

[0042] The cyber security appliance **100** can use the RDA ML module **154** to identify interactive remote desktop activity/session (e.g. a person in a different location than a local computing device can access information stored on and/or access applications on the local computing device over the Internet through software installed on a remote computer and have the ability to perform administrative tasks on the local computing device. For example, the remote access might be able to create new user accounts, change permissions, delete, modify, encrypt or damage sensitive information, shut down or restart the system, etc.) The RDA ML module **154** can identify interactive remote desktop activity by plotting an amount of active connections to data transfer over time and then analyzing the shape of the active connections and data transfer graph with machine learning to identify shapes that are unusual for that organization as well as shapes that match up well with known malicious RDP type attacks, such as ransomware attacks. Based upon parsing the traffic, the DPD manager **152** can intelligently detect the protocol that is being parsed is RDP.

[0043] The cyber security appliance **100** using the RDA ML module **154** can identify remote desktop activity by the shape of the active connections and data transfer along with anomalous use of remote desktop tools in unlikely devices or at unlikely times of a typical workday. Note, that identifying these RDP connections can be difficult with all of the noise of network traffic. The RDA ML module **154** can also distinguish legitimate activity from malicious activity by unauthorized users or malicious attackers via interactive remote desktop activity. For example, an enterprise network might regularly use TeamViewer for normal business but the unexpected presence of more niche tools like Amyy Admin might indicate unauthorized intrusions.

[0044] FIG. **2** illustrates a graph of an embodiment of the RDA ML module identifying when the cyber threat is sending IT network traffic to an external host that is part of an RDP type session by a shape of i) an amount of active connections, ii) an amount of external data transfer, iii) over time, and iv) whether an RDP session would be unusual. The RDA ML module **154** understands that the shape of various cases of active remote desktop tools and other applications seem to have a very characteristic shape that can be relatively easily identified. For example, the basket slash rectangular shape indicated between the two black heroes is indicative an example indicative shape of a remote desktop protocol session. The amount of connections is measured against the amount of external data transfers over time. Low level constant data transfers (e.g. noise) can occur on a regular basis consistently as shown on the graph. Occasional spikes of data transfer to external hosts occur in the network. In addition, a consistent data transfer to an external host over a period of time with a relatively consistent set of data can occur in the shape of a rectangle and/or a basket shape (indicated by the arrows) and this will be indicative of a remote desktop protocol type session period.

[0045] The DPD manager **152** can create a prior profile associated with any Cloud provider, such as AWS, that contains mappings. The DPD manager **152** can build up a basic profile of common mappings that can be shipped by default and allow the user to select to supply as input into the DPD manager **152** depending on each scenario. For example, the DPD manager **152** might observe the same set of non-standard ports in the same public cloud platform, e.g. AWS, so it then make senses to create a prior profile associated with the cloud provider, such as AWS, that contains those mappings. If incorrect, the system already has a mechanism to prune out incorrect mappings. The DPD manager **152** can use a network ingester that can be shipped with these deployment-specific mappings. These deployment-specific mappings can cooperate with a new probe (physical/virtual) or the network ingester. The new probe and/or the network ingester shipped with the deployment-specific mappings allow it to immediately automatically adjust to the specificities of the network environment. Across the fleet of cyber security appliances installed in different network environments. The new probe and/or the network ingester can be fed the collected frequency of non-standard configuration settings that are seen and whether the same non-standard configuration settings in a first network are very similarly set up in a second network. When the same non-standard configuration settings are replicated across multiple different networks in the fleet, then these mappings to any new network traffic consumers are added to the deployment. This can be very useful where the system has autoscaling probes like vSensors in the cloud.

[0046] FIGS. **3***a* and **3***b* illustrate graphs of an embodiment of the RDP ML module in identifying when the cyber threat is using an RDP type session in FIG. **3***a* or not and maybe just a large data upload in FIG. **3***b*. FIG. **3***a* shows a graph of an example data transfer compatible with an interactive remote desktop session. FIG. **3***b* shows a graph of an example data transfer compatible with another service such as Dropbox. The graphs in FIGS. **3***a* and **3***b* represent the characteristic shape of data transfers to (i) endpoints associated with an interactive remote desktop session, for example, TeamViewer (e.g. an example data transfer compatible with an active remote desktop session), and (ii) endpoints associated to other types of services, for example, Dropbox (e.g. an example data transfer compatible with another service such as Dropbox). The graph plots data uploaded to an external host to a number of active connections over time.

[0047] The RDA ML module **154** can first identify the time period around which interactive remote desktop control is anticipated, and then the network data 'signal' is processed by first identifying the time point where data is being sent out of the network. From this point, features of the network data may be extracted and analysed to assess whether interactive remote desktop control/activity is taking place. The following signal characteristics could be used (example values): [0048] 1. Time range: 1 hour [0049] 2. Download/Upload ratio: 0.037 [0050] 3. Total data out: 1.16 GB [0051] 4. Total data in: 41 MB [0052] 5. Out bytes variation fraction (every 5 mins): 0.6% [0053] 6. In bytes variation fraction (every 5 mins): 49.5% [0054] 7. Active conn period/active data transfer period: 0.96 [0055] 8. Rarity: 100% (host), 93% (IP), 40% (domain)

[0056] An alternative set of data signal characteristics would be, for example (example values): [0057] 1. Time range: 1 hour [0058] 2. Download/Upload ratio: 0.98 [0059] 3. Total data outgoing: 112 MB [0060] 4. Total data incoming: 110 MB [0061] 5. Coefficient of variation of incoming data: 0.6% [0062] 6. Coefficient of variation of outgoing data: 0.4% [0063] 7. Coefficient of correlation between incoming and outgoing data: 93% [0064] 8. Duration of active session and duration of transfer ratio: 94% [0065] 9. Rarity of the connected endpoint: 100% [0066] 10. Previous usage of the service by the device: False

[0067] These features can be used in the RDA ML module **154** to train e.g., a random forest AI classifier, which will help to discern if the transfer corresponds to a suspicious remote interactive session or not. Alternatively, the 'signal' could be processed with a neural network (GRU or CNN components) which would be able to identify and classify the network traffic to classify it as remote desktop control or otherwise. A random forest AI classifier can be a machine learning algorithm used for classification tasks, which combines the predictions from multiple decision trees trained on different subsets of data, effectively acting like a “forest” of decision trees to improve the overall accuracy and reduce overfitting by taking a majority vote from each tree's prediction; it's considered a popular and robust method in supervised learning due to its versatility and ability to handle large datasets. The random forest AI classifier can use ensemble learning, where multiple models are combined to make a final prediction, leading to better performance than a single model. Each “tree” in the forest is a decision tree, which makes predictions based on a series of questions about the data. To prevent overfitting, each decision tree is trained on a random sample of the data and a random subset of features, leading to diversity among the trees. Other interesting features can include the rarity of the endpoint, the domain, the IP address, the ASN, etc.

[0068] The RDA ML module **154** may identify interactive remote desktop activity/session of networked devices, and hence identify potential malicious control of that device.

[0069] The RDA ML module **154** can determine if this is likely associated with this activity or not, and whether the shape is the same. The RDA ML module **154** filters the data shape, such as a shape associated with interactive remote desktop activity, and active connections, and it's able to discern if this is an RDP session with a legitimate service, such as Microsoft Remote Desktop, or an interactive remote desktop session with a connection to an unknown remote desktop tool in order to aid in early detection of malicious software of an advanced persistence cyber threat, like ransomware, that has this characteristic when deploying for its cyber-attack.

[0070] The RDA ML module **154** can work with the DPD manager **152** to identify when a cyber threat is using interactive remote desktop activity. For example, if the DPD manager **152** is parsing the IT traffic and cannot tell right off the bat that it is a particular type of RDP like protocol, for example, the malicious ransomware is communicating on some other port than port **3389** (typically associated with RDP traffic), then the RDA ML module **154** could look at the underlying IT traffic and the shape of the active connections and the data transfer to determine that an interactive RDP type session was occurring even if that traffic was not going through the typical port for RDP traffic port **3389**. The RDA ML module **154** can still identify the network traffic as using a type of interactive remote desktop activity like typical RDP protocol because the analyzed traffic has a distinctive shape and detail, to determine affirmatively that the network traffic is using interactive remote desktop activity. The RDA ML module **154** can determine that an interactive remote session is occurring even when the traffic is using not going through port **3389** or using other protocols, e.g., HTTPS, SSH, etc. The RDA ML module **154** can still identify high level properties associated to the interactive remote management/desktop activity without relying on specific application protocol detection such as RDP, etc. even when the network traffic is in encrypted transmissions.

[0071] Next, the RDA ML module **154** can use an AI classifier to analyze the shape of the data-looking at the shape and the number of data transfers and active connections over to identify data transfers in this particular network environment by comparison to known graphs. The RDA ML module **154** can use a graph creation component to create and plot the graph and the combination of the AI classifier and a model with a CNN combined analysis. The RDA ML module **154** has the graph creation component to receive the input data on the IT network traffic in a time series form and transform that into a graph, build up the graph, create the graph, and then analyze that graph to known past shapes associated with interactive remote desktop sessions.

[0072] Next, the RDA ML module **154** assists in investigating serious cybersecurity compromises that start with anomalous and unusual use of interactive remote session/activity tools being used in unlikely devices, or in unlikely scenarios (e.g. unlikely user, unlikely RDP host site, unlikely time of day, etc.) based upon a normal pattern of behaviour for that associated user including hostnames of services that user's uses. For example, a data server of a company is being seen producing connections and data transfers to a hostname associated with a service, such as Team Viewer, at a very unusual time, which is a new service for the corporate environment.

[0073] The RDA ML module **154** understands that from the network traffic perspective, a device being manipulated by a remote desktop tool will show a series of outgoing encrypted connections to a (rare) remote endpoint. Therefore, in practice, timestamps and data volumes may only be available.

[0074] The remote desktop activity (RDA) machine learning module can work with and supplement the DPD manager **152** to assist in identifying when the cyber threat, such as an advanced persistent threat, is using an interactive remote desktop session and a match is not found in a library of services that legitimately provide a type of remote desktop control functionality. The RDA ML module **154** can be configured to detect for an interactive remote desktop session by looking at the shape and the amount of data transfers and active connections over time to identify data transfers in a corporate network environment to distinguish a particular type of RDP sessions compared to other sessions and traffic types, and after identifying the particular type of RDP sessions are present, then determining whether they are legitimate or not.

[0075] A library referenced by the machine learning in the RDA ML module **154** can factor in a hostname going to the Internet and make a compilation of services, for example, 90 services, that legitimately provide this type of remote desktop control and then some subset that are regularly used by this type of user in the network. This method is neutral on that it will use properties of the hostname and pattern of life in a machine learning approach.

[0076] In general, IT networks are busy with lots of legitimate and data transfers and connections. This RDA ML module **154** is able to identify a very specific type of data transfer by looking a very small number of connection properties, data volume, and timing to discern, and check if they are part of a legit host endpoint and/or the same if they are part of a legitimate service; or a suspicious remote interactive session, which might be connected to an early stage serious, illegitimate attack. Thus, the RDA ML module **154** detects whether an interactive remote desktop session is being used by looking at the number of active connections and data traffic over time, and comparing that shape, to see if we can detect an early detection of ransomware or something of that sort.

[0077] In some cases, the destination hostname might be associated with a popular remote desktop tool, for example, b-lon-anx-r021.teamviewer[.]com; and therefore, it will be relatively straightforward to identify them, either by using a list of known endpoints or by training dedicated models. In other cases, the identification using this approach might be more difficult, e.g., a1-server-prod-even.action1[.]com.

[0078] The hostname category classifier can be used to identify if the hostname is part of a well-known legitimate messaging or videoconferencing app, to exclude them. The RDA ML module **154** can include a large language model that will determine if a host is likely, is probably associated with this type of activity or not. The LLM may be an efficiently fine-tuned large language model, in particular using the Low-Rank Adaptation (LoRA) technique. The efficiently fine-tuned large language model will in practice take a very small amount of CPU and memory resources (compared to generally trained LLM) because it is efficiently trained. The RDA ML module **154** can analyze what ports were being looked at and gather the data in a time series form.

[0079] A further step to determine with more precision the nature of the event focuses on the analysis of the destination hostname where the data is being externally sent to. The remote desktop activity (RDA) machine learning module is configured to work with and supplement the DPD manager **152** to assist in identifying when the cyber threat, such as an advanced persistent threat, is uploading IT network traffic to a destination hostname that is part of an interactive remote desktop session by a combination of a data shape analysis of the uploaded IT network traffic and a fine-tuned Large Language Model's analysis of the destination hostname where the data is being externally sent to provide a very accurate method to identify anomalous remote sessions. A fine-tuned Large Language Model (e.g. an LLM using a Low Rank Adaptation) is trained to determine possible categories associated to a given hostname. These categories can include, for example, cloud storage, streaming, etc., and interactive remote desktop endpoints. The fine-tuned Large

Language Model automatically determines with a certain confidence level the possible categories associated with the endpoint. For example, onedrive[.]com and dropbox[.]com would be associated with cloud storage with a score>99%.

[0080] The combination of i) the data shape analysis and ii) the fine-tuned LLM's analysis of the destination hostname where the data is being externally sent to provides a very accurate method to identify anomalous remote sessions.

[0081] Again, the RDA ML module **154** takes advantage of a library making a compilation of a list of known existing remote desktop applications and puts them into categories of legitimate, mostly legitimate, and perhaps shady. The RDA ML module **154** uses a large language model that will determine if a host is likely, is probably associated with this type of activity or not. So, it is not just using whitelists or blacklists.

[0082] The RDA ML module **154** may use an AI classifier to analyze the shape of the data and a language model classifier to investigate the host where this data is being sent. The RDA ML module **154** collects the time series of data blocks going and coming to the endpoint and the hostname. And with this combination of data, the RDA ML module **154** can use the LLM or a couple of AI classifiers to do the prediction of whether a typical RDP session is occurring or an interactive remote desktop session is occurring with a potentially malicious endpoint destination and then feed that as an input as to whether an ADT cyber threat is active in the network under analysis.

[0083] The RDA ML module **154** can train up the AI classifier and the conventional neural networks with supervised machine learning training. The RDA ML module **154** can train with labeled data on different shapes, for example, for the random forest shapes that resemble remote desktop applications and other product processes. The RDA ML module **154** can train to reference the library of host names and destinations to produce a True or False with an associated probability score. Then to particularize for this network under analysis, the language model learns host names that have been known to be used and their frequency as part of remote desktop services in the history of this network.

[0084] Note, Remote Desktop Protocol (RDP) can be a network communication protocol that allows a person to remotely access and control a local computing device over a network from another remote computing device. An interactive remote desktop session essentially enables the person to operate and administrate on the local computing device from a distance as if they were sitting in front of it; it is considered a secure protocol as it establishes an encrypted communication channel between devices.

[0085] The cyber security appliance **100** uses a combination of AI classifiers and machine learning models in the RDA ML module **154** to identify files that have suspicious extensions and/or encrypted files to aid in the detection of an advanced persistent threat, such as ransomware, but identification of these symptoms of an APT may occur to late in an APT cyberattack because these symptoms appear generally when the APT cyberattack is already happening in its attack phase and not earlier in time during the APT's more investigative and stealthy spreading throughout the network phase. Detection of Remote Desktop activities can occur very early in time during the APT's more investigative and stealthy spreading throughout the network phase.

[0086] The RDA ML module **154** identifies remote desktop control of a networked device, and hence identifies potential malicious control of that device.

[0087] Several key takeaways can be as follows. The most important is that it is an early detection of potentially very serious attacks in the very early stages, makes things much easier to mitigate. The second takeaway is that the RDA ML module **154** can be implemented with machine learning technology and cyber security technologies that are efficient in computational terms. The RDA ML module **154** can arrive at a verdict of whether the network traffic of data uploaded to an external host is part of a remote desktop type session or some other sort of external data transfer in milliseconds.

Additional Details

[0088] 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.

[0089] FIG. **4** illustrates a block diagram of an embodiment of the AI-based cyber security with example components, including the DPD manager and the RDA ML module, 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**, a DPD manager **152**, a RDA ML module **154**, 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 simulator **105**, and other similar components.

[0090] 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**.

[0091] 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**.

[0092] 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.

[0093] 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 FIG. **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.

[0094] 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 FIG. **9**). 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 FIG. **9**).

[0095] The artificial intelligence-based cyber security analyst tool can use the cyber threat analyst module **120** and its interaction with the other modules and AI models **160** in the cyber security appliance **100**. The artificial intelligence-based cyber security analyst tool's investigations into potential cyber-attacks from potential cyber threats has the ability for customers to review the outcomes of the artificial intelligence-based cyber security analyst tool's investigations at the hypothesis-level (hypothesis steps taken and investigation steps taken and then its conclusion) and a human readable summary on why the system took the hypothesis steps taken and investigation steps taken and then its conclusion. For every compatible DETECT alert (e.g., model breach), the artificial intelligence-based cyber security analyst tool investigates a series of possible relevant hypotheses and we try and find data and find activity that meets the criteria, those hypotheses. When artificial intelligence-based cyber security analyst tool finds activity and/or data that meets the criteria that actually support a likelihood of a hypothesis, then those hypotheses are worth surfacing to an operator. The artificial intelligence-based cyber security analyst tool presents the most salient information to the end user. The artificial intelligence-based cyber security analyst autonomously investigates alerts, streamlines investigations and prioritizes incidents, thus reducing workload and alert fatigue. The artificial intelligence-based cyber security analyst uses various forms of machine learning, including unsupervised, supervised, and deep learning combined with human intuition and trade craft from hundreds of world-class human cyber analysts across thousands of customer deployments. The artificial intelligence-based cyber security analyst relieves a human cyber analyst from spend anywhere between half an hour and half a day investigating a single suspicious security incident. The artificial intelligence-based cyber security analyst looks for patterns, forms hypotheses, reaches conclusions about how to mitigate the threat, and shares the findings with the rest of the business. The artificial intelligence-based cyber security analyst continuously conducts investigations behind the scenes and operating at a speed and scale beyond human capabilities. The artificial intelligence-based cyber security analyst tool as a large language model (LLM) is built to incorporate cyber threat knowledge from external data stores, external data sources, as well as from a network's own cyber security appliance. The artificial intelligence-based cyber security analyst tool uses threat intelligence to understand a cyber threat adversary tactics and motivations. An effectiveness of the artificial intelligence-based cyber security analyst tool lies in its ability to access and integrate diverse data sources. The artificial intelligence-based cyber security analyst can tap into external data stores, such as threat intelligence platforms and vulnerability databases, to enrich its understanding of the threat landscape.

[0096] 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 FIG. **9**, 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.

[0097] Returning back to FIG. **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, FIG. **9** and a chain of linked low-level indicators, including abnormal behavior compared to the normal pattern 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.

[0098] 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.

[0099] 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.

[0100] 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**).

[0101] 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.

[0102] 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.

[0103] 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.

[0104] 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. FIG. **4** shows the example components making up the detection engine to include interfaces to the cyber-attack simulator **105**, the autonomous response engine **140**, and the restoration engine **190**.

[0105] 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.

[0106] FIG. **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 simulator **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.

[0107] Referring to FIG. **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.

[0108] 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 simulator **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 FIG. **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 simulator **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, FIG. **6**)

[0109] 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.

[0110] 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.

[0111] 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.

[0112] 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.

[0113] 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.

[0114] In an example, the autonomous response engine **140** uses its intelligence to cooperate with a cyber-attack simulator 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.

[0115] In another example, the cyber-attack simulator **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.

[0116] 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**.

[0117] FIG. **6** illustrates a block diagram of an embodiment of the cyber-attack simulator with Artificial Intelligence-based simulations conducted in the cyber-attack simulator 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 FIG. **6**, the various cooperating modules residing in the cyber-attack simulator **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.

[0118] The cyber-attack simulator **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 simulator **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 simulator **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 simulator **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 FIGS. **7**A and **7**B.

[0119] FIG. **7**A illustrates a diagram of an embodiment of the cyber-attack simulator 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 simulator **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.

[0120] 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.

[0121] FIG. **7**B illustrates a diagram of an embodiment of the cyber-attack simulator 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.

Training of AI Pre-Deployment and then During Deployment

[0122] 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.

[0123] 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.

[0124] 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.

[0125] 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.

Clustering Methods

[0126] 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.

[0127] 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.

[0128] 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.

[0129] 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.

Ranking the Cyber Threat

[0130] The assessment module **125** and/or cyber threat analyst module **120** of FIG. **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.

More on the Operation of the Cyber Security Appliance

[0131] 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.

[0132] 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.

Anomaly Detection/Deviations

[0133] 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.

[0134] 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 is 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.

[0135] 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 U.S. Pat. No. 10,701,093 granted Jun. 30, 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 Feb. 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 Apr. 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.

[0136] 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.

Additional Module Interactions

[0137] Referring to FIG. **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.

[0138] 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.

[0139] 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.

[0140] 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.

[0141] 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.

Determination of Whether Something is Likely Malicious

[0142] 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.

[0143] 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.

[0144] 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.

[0145] 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.

[0146] 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.

[0147] 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.

[0148] 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.

[0149] 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.

[0150] 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.

[0151] 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?

[0152] 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.

[0153] 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.

[0154] 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. FIG. **9** 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.

[0155] 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.

An Assessment of the Cyber Threat in Order to Determine Appropriate Autonomous Actions, for Example, Those by the Autonomous Response Engine

[0156] 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.

[0157] 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.

[0158] 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.

[0159] 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.

[0160] 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.

[0161] 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.

[0162] 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.

[0163] 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.

[0164] 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.

[0165] 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.

[0166] 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.

[0167] 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, Remove VM permissions, modify networking rules, Modify IAM user or role policies, 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.

[0168] 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.

[0169] 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.

[0170] 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.

[0171] 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.

[0172] 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.

[0173] Referring to FIG. **6**, the cyber-attack simulator **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 **791**A-B, and an IT network system **792** with one or more entities, over one or more networks **791**/**792** in the system being protected.

[0174] The cyber-attack simulator **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**.

[0175] The cyber-attack simulator **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.

[0176] 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 simulator **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.

[0177] The node exposure score generator and the simulated attack module **750** in the cyber-attack simulator **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 FIGS. **7**A and **7**B. 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 simulator **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 simulator **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.

[0178] The cyber-attack simulator **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 simulator **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.

[0179] 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.

[0180] The cyber-attack simulator **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.

[0181] 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 simulator **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.

[0182] 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.

[0183] The cyber-attack simulator **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 simulator **105** with Artificial Intelligence-based simulations in the cyber-attack simulator **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 simulator cooperates with the AI models modelling a normal pattern of life for entities/nodes in the system being protected.

[0184] 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.

[0185] 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.

[0186] FIG. **8** illustrates a block diagram of an embodiment of the AI-based cyber security appliance **100** 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.

[0187] Referring back to FIG. **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.

[0188] 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.

[0189] 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.

[0190] 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.

[0191] 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**.

[0192] 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.

[0193] 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.

[0194] 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.

[0195] 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.

[0196] 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.

[0197] 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.

[0198] 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.

[0199] 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.

[0200] 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.

Computing Devices

[0201] FIG. **10** 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 discussed herein.

[0202] 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.

[0203] 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**.

[0204] 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**.

[0205] 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.

[0206] 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.

[0207] 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 can be 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. The systems and methods described herein can be implemented with these algorithms discussed herein.

[0208] 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.

[0209] 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 appliance to detect a cyber threat, comprising: a Deep Packet Detection (DPD) manager configured to adaptively parse network traffic with a DPD machine learning (ML) engine based upon determining i) a port configuration setting in a network server in a network and ii) a protocol utilized by the network traffic, under analysis, where the DPD manager is further configured to be capable of detecting 1) a non-standard configuration set up for the network traffic to be processed by a port on the network server, 2) a non-standard protocol utilized by the network traffic, and 3) any combination of both, and then complete a deep packet inspection upon the network traffic that has 1) the non-standard configuration set up for the network traffic to be processed by the port on the network server, and/or 2) the non-standard protocol utilized by the network traffic; and where instructions implemented in software for the DPD manager and the DPD ML engine are configured to be stored in one or more non-transitory storage mediums to be executed by one or more processing units.

2. The cyber security appliance of claim 1, where the network traffic under analysis is IT network traffic and the network is an IT network, where the DPD manager is further configured to not have to make assumptions about i) the port configuration setting in the network server in the IT network or ii) the protocol utilized by the IT network traffic, under analysis, but rather have the DPD ML engine learn after being deployed with unsupervised machine learning but over time about which particular ports are servicing the network traffic in this IT network and what protocols are being utilized by the IT network traffic in this IT network, and wherein the DPD manager is configured to then make a mapping of the particular ports that are servicing the network traffic in this IT network and what protocols are being utilized by the IT network traffic in this IT network.

3. The cyber security appliance of claim 1, where the DPD manager is further configured to create a matrix in a memory of the cyber security appliance of all of the ports servicing network traffic in the network and then to analyze for and store in the memory metadata associated with network traffic being processed by each port, which is then fed to the DPD ML engine to deduce and predict what type of network traffic is being process by each port for the network.

4. The cyber security appliance of claim 1, where the network traffic under analysis is IT network traffic and the network is an IT network, and where the DPD manager is further configured to I) cooperate and receive input from network sensors configured to perform deep packet inspection upon the IT network traffic in order to detect and determine 1) a standard configuration set up for IT network traffic to be processed by the port on the network server and a standard IT protocol utilized by the IT network traffic as well as II) cooperate and feed in meta data to the DPD ML engine to determine 1) the non-standard configuration set up for IT network traffic to be processed by the port on the network server, 2) the non-standard IT protocol utilized by the IT network traffic, and 3) any combination of both.

5. The cyber security appliance of claim 1, where the DPD manager is configured to use a protocol analyzer to check the non-standard protocol utilized by the network traffic against a library of known protocols for the network traffic and when no match occurs then to feed metadata and at least partially recognized characteristics of a given known protocol over to the DPD ML engine to deduce and predict what type of protocol is being used by the network traffic.

6. The cyber security appliance of claim 1, where the network traffic under analysis is IT network traffic and the network is an IT network, and where the DPD manager is configured to use a port state component to store a relationship of the non-standard protocol utilized by the IT network traffic, under analysis, to a deduced protocol being used by the IT network traffic by the DPD ML engine and an associated port being used by the IT network traffic using the non-standard IT protocol, and when a subsequent packet in network traffic is seen on the port, then the DPD manager can subsequently rely upon the port state component to provide the deduced protocol being used by the IT network traffic and the associated port being used by the IT network traffic.

7. The cyber security appliance of claim 1, where a remote desktop activity (RDA) machine learning module is configured to work with and supplement the DPD manager to assist in identifying when the cyber threat is sending the traffic to an external host that is part of an interactive remote desktop session by a shape of i) active connections, ii) data transfer, iii) over time and iv) whether an RDP session would be unusual.

8. The cyber security appliance of claim 1, where an RDA machine learning module is configured to work with and supplement the DPD manager to assist in identifying when the cyber threat is uploading IT network traffic to a destination hostname that is part of an interactive remote desktop session by a combination of a data shape analysis of the uploaded IT network traffic and a Large Language Model's analysis of the destination hostname where the data is being externally sent.

9. The cyber security appliance of claim 1, where an RDA machine learning module is configured to work with and supplement the DPD manager to assist in identifying when the cyber threat is using an interactive remote desktop session and a match is not found in a library of services that legitimately provide a type of remote desktop control functionality.

10. A method for a cyber security appliance to detect a cyber threat, comprising: providing a Deep Packet Detection (DPD) manager to adaptively parse network traffic with a DPD machine learning (ML) engine based upon determining i) a port configuration setting in a network server in a network and ii) a protocol utilized by the network traffic, under analysis; and providing the DPD manager to be capable of detecting 1) a non-standard configuration set up for the network traffic to be processed by a port on the network server, 2) a non-standard protocol utilized by the network traffic, and 3) any combination of both, and then complete a deep packet inspection on the network traffic that has 1) the non-standard configuration set up for the network traffic to be processed by the port on the network server, and/or 2) the non-standard protocol utilized by the network traffic.

11. The method for the cyber security appliance of claim 10, further comprising: where the network traffic under analysis is IT network traffic and the network is an IT network, and providing the DPD manager to not have to make assumptions about i) the port configuration setting in the network server in the IT network or ii) the protocol utilized by the IT network traffic, under analysis, but rather have the DPD ML engine learn after being deployed with unsupervised machine learning but over time about which particular ports are servicing the network traffic in this IT network and what protocols are being utilized by the IT network traffic in this IT network; and providing the DPD manager to then make a mapping of the particular ports that are servicing the network traffic in this IT network and what protocols are being utilized by the IT network traffic in this IT network.

12. The method for the cyber security appliance of claim 10, further comprising: providing the DPD manager to create a matrix in a memory of the cyber security appliance of all of the ports servicing network traffic in the network and then to analyze for and store in the memory metadata associated with network traffic being processed by each port, which is then fed to the DPD ML engine to deduce and predict what type of network traffic is being process by each port for the network.

13. The method for the cyber security appliance of claim 10, further comprising: where the network traffic under analysis is IT network traffic and the network is an IT network, and providing the DPD manager to I) cooperate and receive input from network sensors configured to perform deep packet inspection upon the IT network traffic in order to detect and determine 1) a standard configuration set up for the IT network traffic to be processed by the port on the IT network server and a standard IT protocol utilized by the IT network traffic as well as II) cooperate and feed in meta data to the DPD ML engine to determine 1) the non-standard configuration set up for IT network traffic to be processed by the port on the IT network server, 2) the non-standard IT protocol utilized by the IT network traffic, and 3) any combination of both.

14. The method for the cyber security appliance of claim 10, further comprising: providing the DPD manager is configured to use a protocol analyzer to check the non-standard protocol utilized by the network traffic against a library of known protocols for the network traffic and when no match occurs then to feed metadata and at least partially recognized characteristics of a given known protocol over to the DPD ML engine to deduce and predict what type of protocol is being used by the network traffic.

15. The method for the cyber security appliance of claim 10, further comprising: where the network traffic under analysis is IT network traffic and the network is an IT network, and providing the DPD manager to use a port state component to store a relationship of the non-standard protocol utilized by the IT network traffic, under analysis, to a deduced protocol being used by the IT network traffic by the DPD ML engine and an associated port being used by the IT network traffic using the non-standard IT protocol, and when a subsequent packet in network traffic is seen on the port, then the DPD manager can subsequently rely upon the port state component to provide the deduced protocol being used by the IT network traffic and the associated port being used by the IT network traffic.

16. The method for the cyber security appliance of claim 10, further comprising: providing a remote desktop activity (RDA) machine learning module to work with and supplement the DPD manager to assist in identifying when the cyber threat is sending the network traffic to an external host that is part of an interactive remote desktop session by a shape of active connections, data transfer, and whether an RDP session would be unusual.

17. The method for the cyber security appliance of claim 10, further comprising: providing a remote desktop activity (RDA) machine learning module to work with and supplement the DPD manager to assist in identifying when the cyber threat is uploading IT network traffic to a destination hostname that is part of an interactive remote desktop session by a combination of a data shape analysis of the uploaded IT network traffic and a Large Language Model's analysis of the destination hostname where the data is being externally sent.

18. The method for the cyber security appliance of claim 10, further comprising: providing a remote desktop activity (RDA) machine learning module to work with and supplement the DPD manager to assist in identifying when the cyber threat is using an interactive remote desktop session, and a match is not found in a library of services that legitimately provide a type of remote desktop control functionality.

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 a Deep Packet Detection (DPD) manager in a cyber security appliance to adaptively parse Information Technology (IT) network traffic with a DPD machine learning (ML) engine based upon determining i) a port configuration setting in a network server in an IT network and ii) a protocol utilized by IT network traffic, under analysis; and using the DPD manager to be capable of detecting 1) a non-standard configuration set up for IT network traffic to be processed by a port on the network server, 2) a non-standard protocol utilized by the IT network traffic, and 3) any combination of both, and then complete a deep packet inspection on the IT network traffic that has 1) the non-standard configuration set up for IT network traffic to be processed by the port on the network server, and/or 2) the non-standard protocol utilized by the IT network traffic.

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 a remote desktop activity (RDA) machine learning module to work with and supplement the DPD manager to assist in identifying when a cyber threat is uploading IT network traffic to a destination hostname that is part of an interactive remote desktop session by a combination of a data shape analysis of the uploaded IT network traffic and a Large Language Model's analysis of the destination hostname where the data is being externally sent.