US 20250258912A1

(19) United States

(12) Patent Application Publication

YARON et al.

(10) Pub. No.: US 2025/0258912 A1

(43) Pub. Date: Aug. 14, 2025

(54) TECHNIQUES FOR CROSS-SOURCE ALERT PRIORITIZATION AND REMEDIATION

(71) Applicant: **Dazz, Inc.**, San Francisco, CA (US)

(72) Inventors: **Eshel YARON**, Amsterdam (NL); **Barak BERCOVITZ**, Even-Yehuda (IL)

(73) Assignee: **Dazz, Inc.**, San Francisco, CA (US)

(21) Appl. No.: **19/169,674**

(22) Filed: **Apr. 3, 2025**

Related U.S. Application Data

(63) Continuation of application No. 18/962,112, filed on Nov. 27, 2024.

(60) Provisional application No. 63/603,949, filed on Nov. 29, 2023.

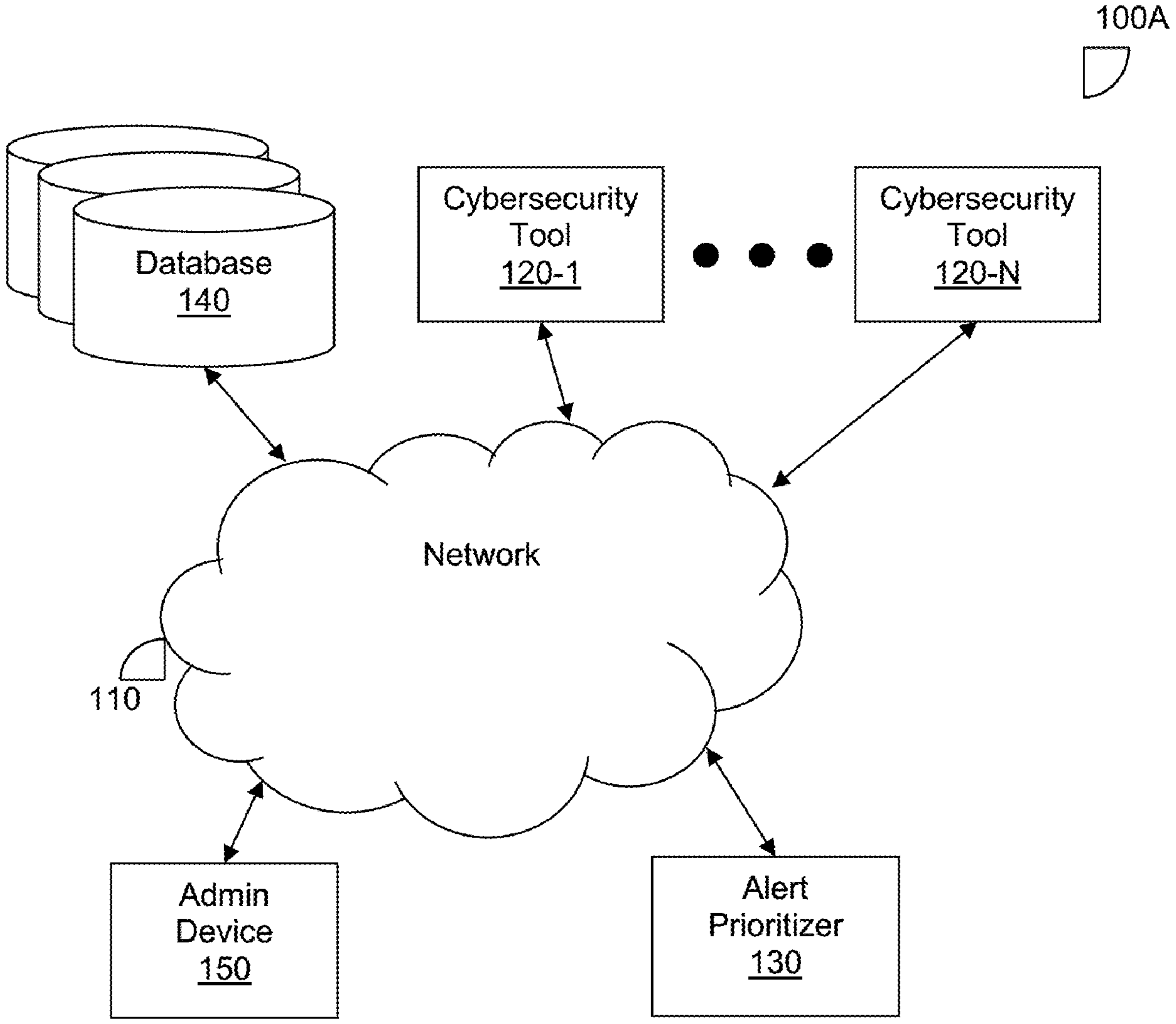
Publication Classification

(51) **Int. Cl.**
G06F 21/55 (2013.01)
G06N 20/00 (2019.01)

(52) **U.S. Cl.**
CPC ***G06F 21/554*** (2013.01); ***G06N 20/00*** (2019.01); *G06F 2221/034* (2013.01)

(57) ABSTRACT

A system and method for machine learning training. A method includes clustering cybersecurity alerts from different alert sources into clusters based on features distances between alerts. A subset of the cybersecurity alerts is selected from each cluster based on the cluster center of the cluster. Each subset is presented via a user device in order to obtain a set of user inputs with respect to each presented subset. A training set is created based on the user inputs with respect to each presented subset. The training set includes the cybersecurity alerts and corresponding training prioritizations. A prioritization machine learning model is trained to prioritize cybersecurity alerts from the plurality of alert sources using the training set.



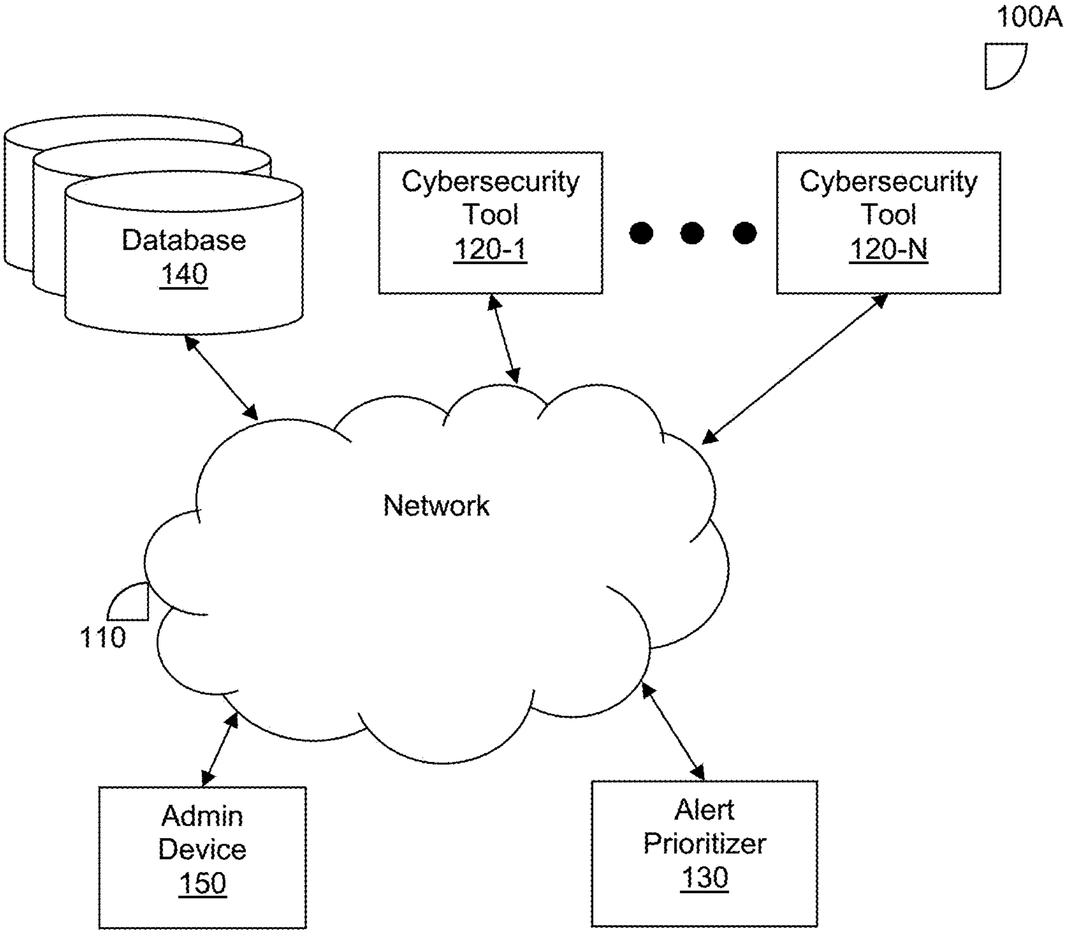

100A
Database
140
Cybersecurity
Tool
120-1
Cybersecurity
Tool
120-N
Network
110
Admin
Device
150
Alert
Prioritizer
130

FIG. 1A

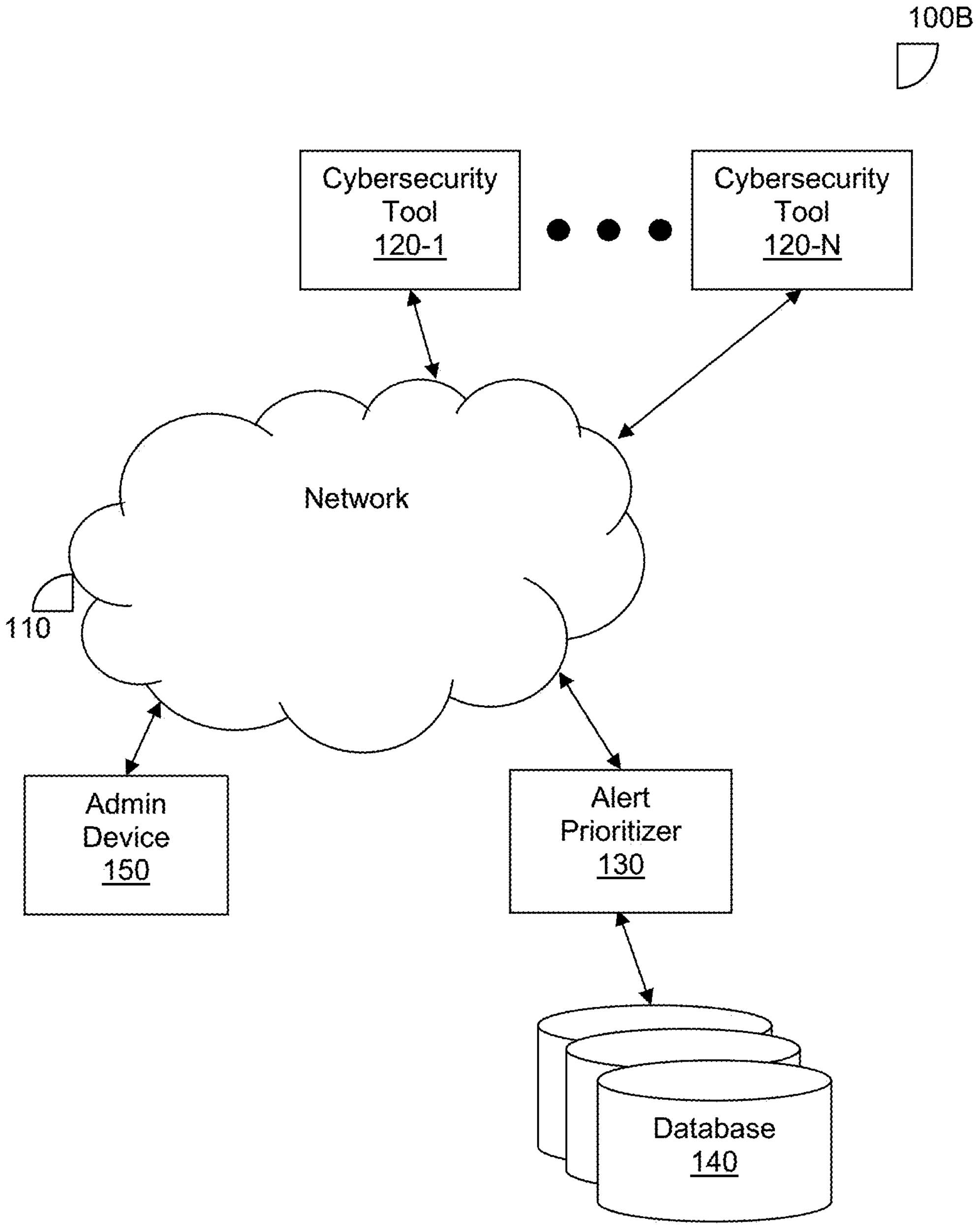

100B
Cybersecurity Tool 120-1
Cybersecurity Tool 120-N
Network
110
Admin Device 150
Alert Prioritizer 130
Database 140

FIG. 1B

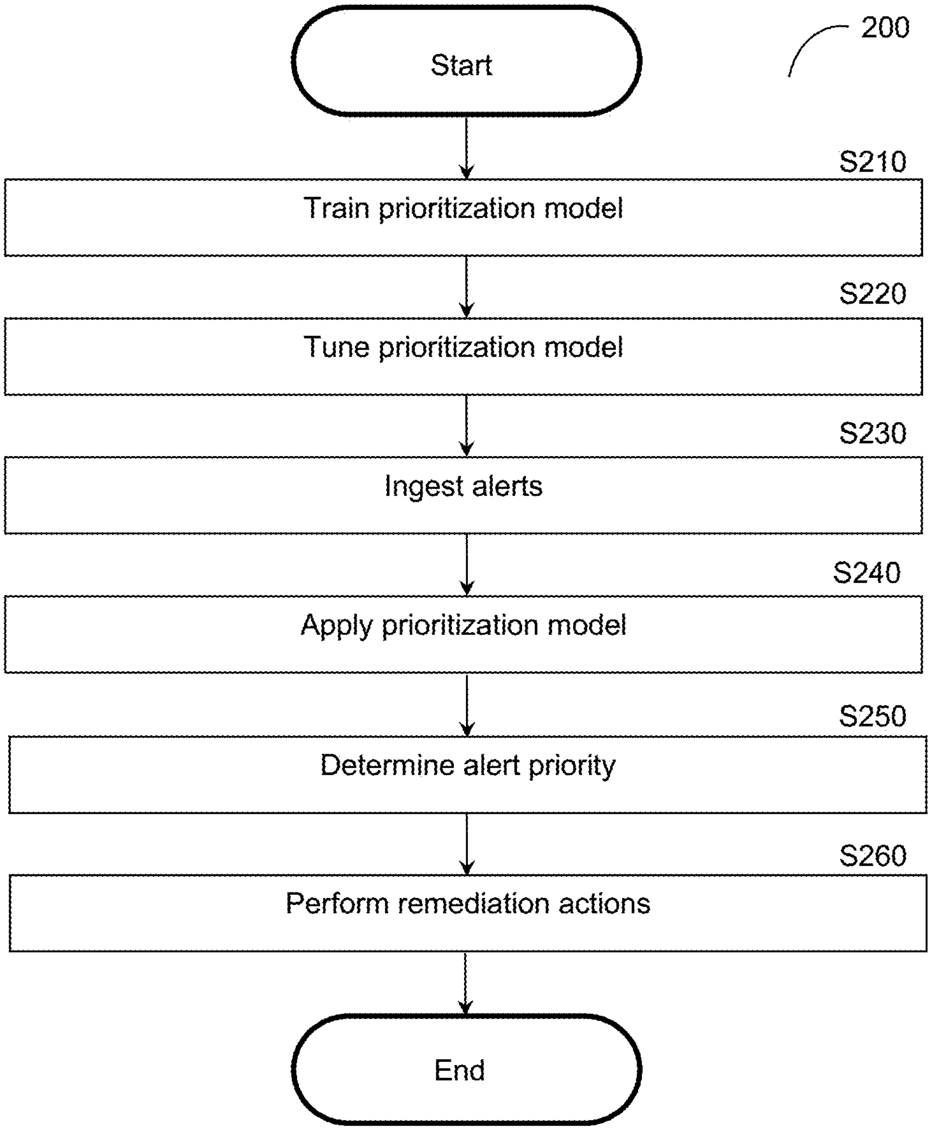

200
Start
S210
Train prioritization model
S220
Tune prioritization model
S230
Ingest alerts
S240
Apply prioritization model
S250
Determine alert priority
S260
Perform remediation actions
End

FIG. 2

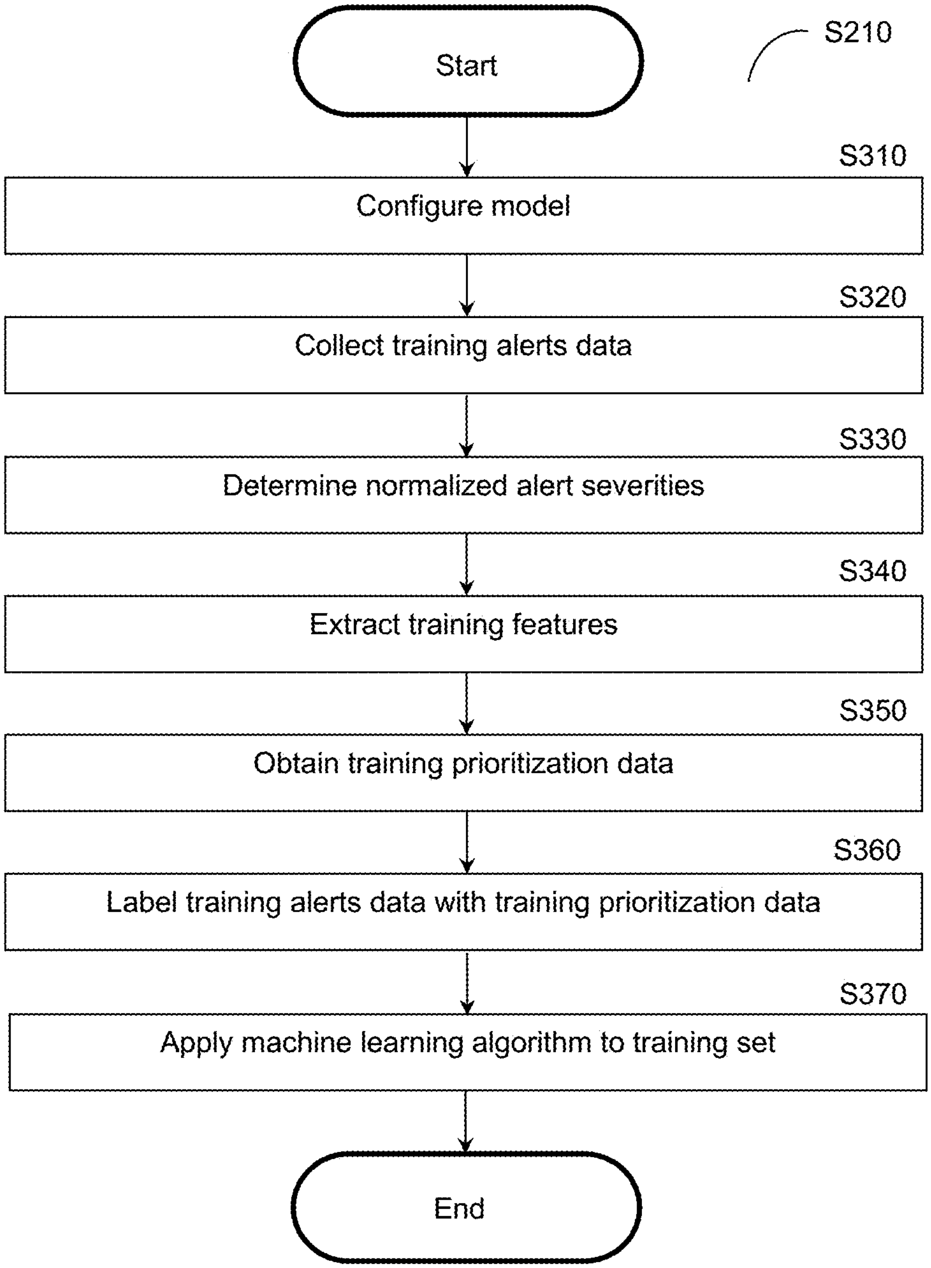

S210
Start
S310
Configure model
S320
Collect training alerts data
S330
Determine normalized alert severities
S340
Extract training features
S350
Obtain training prioritization data
S360
Label training alerts data with training prioritization data
S370
Apply machine learning algorithm to training set
End

FIG. 3

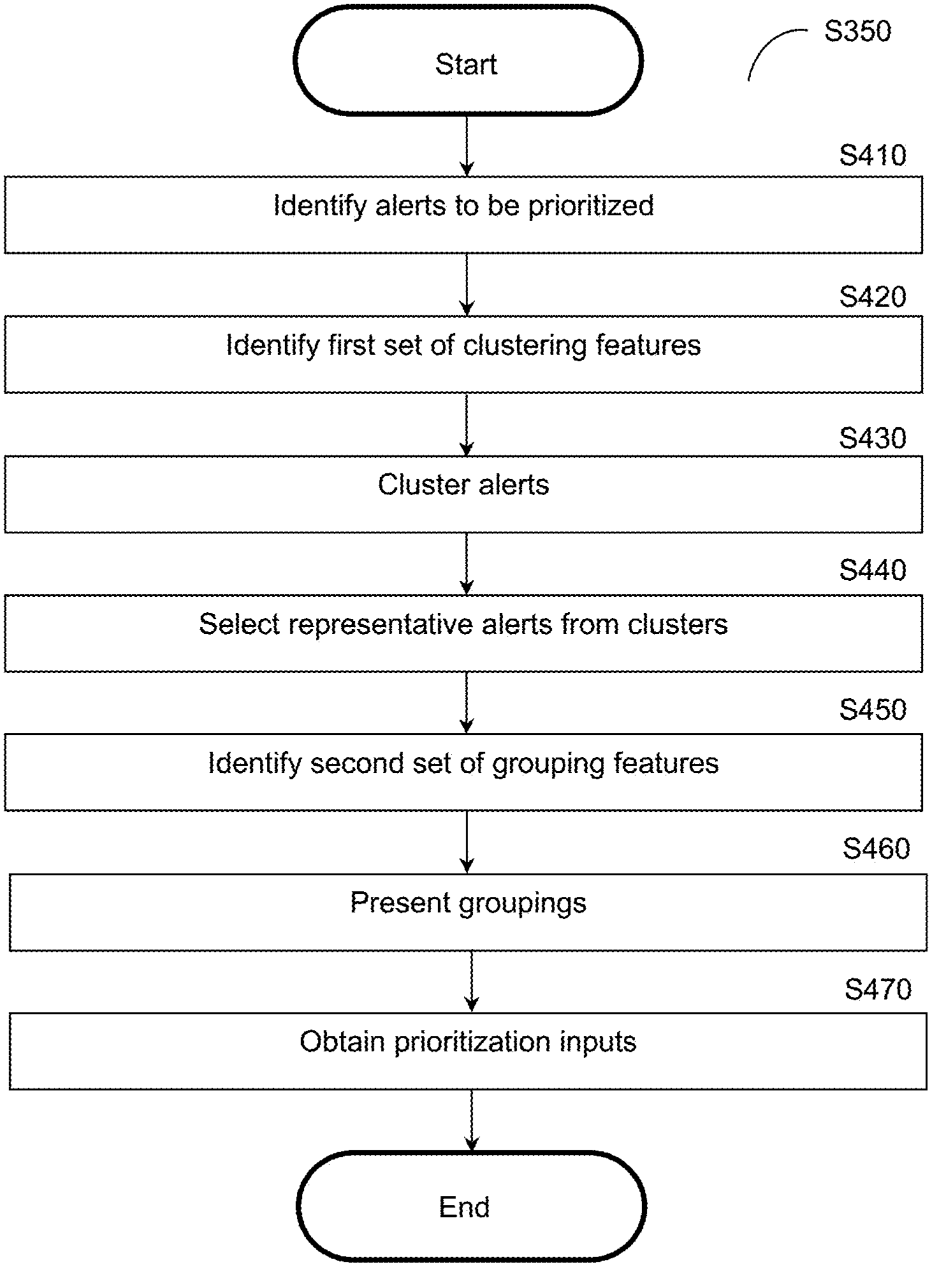

S350
Start
S410
Identify alerts to be prioritized
S420
Identify first set of clustering features
S430
Cluster alerts
S440
Select representative alerts from clusters
S450
Identify second set of grouping features
S460
Present groupings
S470
Obtain prioritization inputs
End

FIG. 4

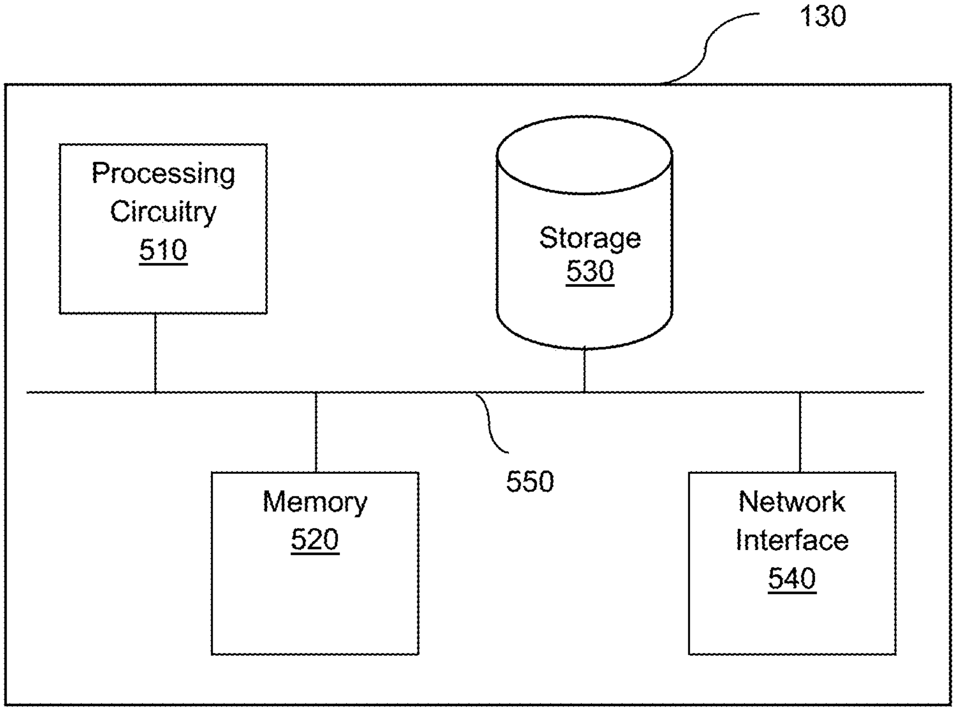

130
Processing Circuitry 510
Storage 530
550
Memory 520
Network Interface 540

FIG. 5

TECHNIQUES FOR CROSS-SOURCE ALERT PRIORITIZATION AND REMEDIATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. patent application Ser. No. 18/962,112 filed on Nov. 27, 2024, now pending, which claims the benefit of U.S. Provisional Patent Application No. 63/603,949 filed on Nov. 29, 2023.

[0002] The contents of the above-referenced applications are hereby incorporated by reference.

TECHNICAL FIELD

[0003] The present disclosure relates generally to cybersecurity alert prioritization, and more specifically to prioritizing alerts generated across cybersecurity tools.

BACKGROUND

[0004] Cybersecurity in modern computing environments, and in particular cloud computing environments, often relies on many different cybersecurity tools for alerting on different potential cyber threats. As more and more specialized tools are developed, organizations become increasingly likely to use multiple different tools in order to manage security in their environments.

[0005] It would therefore be advantageous to provide a solution that would aid in managing alerts across different sources such as different cybersecurity tools.

SUMMARY

[0006] A summary of several example embodiments of the disclosure follows. This summary is provided for the convenience of the reader to provide a basic understanding of such embodiments and does not wholly define the breadth of the disclosure. This summary is not an extensive overview of all contemplated embodiments, and is intended to neither identify key or critical elements of all embodiments nor to delineate the scope of any or all aspects. Its sole purpose is to present some concepts of one or more embodiments in a simplified form as a prelude to the more detailed description that is presented later. For convenience, the term "some embodiments" or "certain embodiments" may be used herein to refer to a single embodiment or multiple embodiments of the disclosure.

[0007] Certain embodiments disclosed herein include a method for machine learning training. The method comprises: clustering a plurality of cybersecurity alerts into a plurality of clusters based on a features distance between alerts among the plurality of cybersecurity alerts, wherein the plurality of cybersecurity alerts is from a plurality of alert sources; selecting a subset of the plurality of cybersecurity alerts from each of the plurality of clusters, wherein each cluster has a cluster center, wherein the subset of the plurality of cybersecurity alerts is selected from each of the plurality of clusters based on the cluster center of the cluster; causing presentation of each subset of the plurality of cybersecurity alerts selected from each of the plurality of clusters via a user device in order to obtain a set of user inputs with respect to each presented subset of the plurality of cybersecurity alerts; creating a training set based on the user inputs with respect to each presented subset of the plurality of cybersecurity alerts, wherein the training set includes the plurality of cybersecurity alerts and a corresponding plurality of training prioritizations for the plurality of cybersecurity alerts; and training a prioritization machine learning model to prioritize cybersecurity alerts from the plurality of alert sources using the training set.

[0008] Certain embodiments disclosed herein also include a non-transitory computer readable medium having stored thereon causing a processing circuitry to execute a process, the process comprising: clustering a plurality of cybersecurity alerts into a plurality of clusters based on a features distance between alerts among the plurality of cybersecurity alerts, wherein the plurality of cybersecurity alerts is from a plurality of alert sources; selecting a subset of the plurality of cybersecurity alerts from each of the plurality of clusters, wherein each cluster has a cluster center, wherein the subset of the plurality of cybersecurity alerts is selected from each of the plurality of clusters based on the cluster center of the cluster; causing presentation of each subset of the plurality of cybersecurity alerts selected from each of the plurality of clusters via a user device in order to obtain a set of user inputs with respect to each presented subset of the plurality of cybersecurity alerts; creating a training set based on the user inputs with respect to each presented subset of the plurality of cybersecurity alerts, wherein the training set includes the plurality of cybersecurity alerts and a corresponding plurality of training prioritizations for the plurality of cybersecurity alerts; and training a prioritization machine learning model to prioritize cybersecurity alerts from the plurality of alert sources using the training set.

[0009] Certain embodiments disclosed herein also include a system for alert prioritization. The system comprises: a processing circuitry; and a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to: cluster a plurality of cybersecurity alerts into a plurality of clusters based on a features distance between alerts among the plurality of cybersecurity alerts, wherein the plurality of cybersecurity alerts is from a plurality of alert sources; select a subset of the plurality of cybersecurity alerts from each of the plurality of clusters, wherein each cluster has a cluster center, wherein the subset of the plurality of cybersecurity alerts is selected from each of the plurality of clusters based on the cluster center of the cluster; cause presentation of each subset of the plurality of cybersecurity alerts selected from each of the plurality of clusters via a user device in order to obtain a set of user inputs with respect to each presented subset of the plurality of cybersecurity alerts; create a training set based on the user inputs with respect to each presented subset of the plurality of cybersecurity alerts, wherein the training set includes the plurality of cybersecurity alerts and a corresponding plurality of training prioritizations for the plurality of cybersecurity alerts; and train a prioritization machine learning model to prioritize cybersecurity alerts from the plurality of alert sources using the training set.

[0010] Certain embodiments disclosed herein include a method, non-transitory computer-readable medium, or system as noted above or below, wherein the plurality of cybersecurity alerts is a plurality of first cybersecurity alerts, further including or being configured to perform the following step or steps: applying the prioritization machine learning model with respect to a plurality of second cybersecurity alerts from the plurality of alert sources in order to obtain a set of outputs; and determining a priority for the plurality of second cybersecurity alerts based on the set of outputs.

[0011] Certain embodiments disclosed herein include a method, non-transitory computer-readable medium, or system as noted above or below, further including or being configured to perform the following step or steps: performing a plurality of remediation actions based on the determined priority.

[0012] Certain embodiments disclosed herein include a method, non-transitory computer-readable medium, or system as noted above or below, wherein each cybersecurity alert among the subset of each grouping belongs to a different cluster of the plurality of clusters than each other cybersecurity alert among the subset of the grouping.

[0013] Certain embodiments disclosed herein include a method, non-transitory computer-readable medium, or system as noted above or below, further including or being configured to perform the following step or steps: determining a plurality of groupings based on the plurality of clusters, wherein the subset of the plurality of cybersecurity alerts selected from each cluster is determined based on the plurality of groupings.

[0014] Certain embodiments disclosed herein include a method, non-transitory computer-readable medium, or system as noted above or below, wherein each of the plurality of cybersecurity alerts has a features vector, wherein each cluster center is a features vector determined based on the features vectors of the plurality of cybersecurity alerts.

[0015] Certain embodiments disclosed herein include a method, non-transitory computer-readable medium, or system as noted above or below, wherein each cluster center includes any of: mean values for a portion of the plurality of cybersecurity alerts, and median values for a portion of the plurality of cybersecurity alerts.

[0016] Certain embodiments disclosed herein include a method, non-transitory computer-readable medium, or system as noted above or below, wherein each training prioritization is defined relative to at least one other training prioritization of the plurality of training prioritizations from at least one different alert source of the plurality of alert sources.

[0017] Certain embodiments disclosed herein include a method, non-transitory computer-readable medium, or system as noted above or below, further including or being configured to perform the following step or steps: determining a plurality of features vectors for the plurality of cybersecurity alerts; and determining a plurality of centers of the plurality of clusters, wherein each features distance is between a features vector for one of the plurality of cybersecurity alerts and a features vector for one of the centers of one of the clusters among the plurality of clusters to which the cybersecurity alert belongs.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The subject matter disclosed herein is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the disclosed embodiments will be apparent from the following detailed description taken in conjunction with the accompanying drawings.

[0019] FIGS. **1**A-B are network diagrams utilized to describe various disclosed embodiments.

[0020] FIG. **2** is a flowchart illustrating a method for alert prioritization and remediation according to an embodiment.

[0021] FIG. **3** is a flowchart illustrating a method for training a prioritization model according to an embodiment.

[0022] FIG. **4** is a flowchart illustrating a method for obtaining prioritization data for alerts according to an embodiment.

[0023] FIG. **5** is a schematic diagram of an alert prioritizer according to an embodiment.

DETAILED DESCRIPTION

[0024] In light of the use of various cybersecurity tools noted above, it has been identified that prioritizing alerts across different sources (e.g., different cybersecurity tools) presents a challenge. More specifically, alerts from different sources may not express urgency in the same manner, and the different sources may determine severities differently. For example, a high severity alert from one source and a high severity alert from another source may not actually represent the same degree of threat. Accordingly, it has been further identified that alerts from different sources cannot be compared directly, and that alerts from different sources cannot be accurately prioritized based on direct comparisons of the explicit information in those alerts. To this end, various disclosed embodiments provide techniques for cross-source alert prioritization that allow for effectively comparing alerts from different sources in order to determine which of those alerts to prioritize over others.

[0025] The various disclosed embodiments include methods and systems for prioritizing alerts and for automating remediation using such prioritizations. In particular, various disclosed embodiments may be utilized to prioritize alerts from different sources (e.g., alerts generated by different cybersecurity tools) in order to provide a cross-source prioritization which may be utilized to more effectively prioritize alerts in spite of any differences in formatting, severity criteria, and other factors which render direct comparisons between alerts from such different sources ineffective. To this end, various disclosed embodiments leverage machine learning techniques in order to learn prioritization tendencies which may be utilized to automate prioritization in a more context-rich manner than existing automated solutions without requiring a user to explicitly define a prioritization procedure.

[0026] In various embodiments, alerts from cybersecurity tools are ingested and prioritized by an alert prioritizer system. The alert prioritizer system includes or is otherwise configured to utilize a prioritization model. In accordance with various disclosed embodiments, the prioritization model is a machine learning model trained based on training data including features of historical alerts and prioritization data for those alerts. The prioritized alerts may be utilized to automatically remediate alerts or otherwise may be presented to a user (e.g., a security systems administrator via an admin device operated by such an administrator) to assist in remediation efforts.

[0027] In this regard, it is noted that remediation is typically an expensive process in terms of resources, whether computer resources or human resources. Optimizing prioritization using the various techniques described herein may aid in reducing or otherwise optimizing use of resources, particularly computing resources, in order to more efficiently remediate potential cyber threats. Additionally, correctly prioritizing alerts allows for mitigating potential cyber threats faster than if alerts are not correctly prioritized.

[0028] In an embodiment, the prioritization model is trained using a machine learning algorithm, such as, but not limited to, a genetic algorithm or simulated annealing algorithm. In some embodiments, the prioritization model is trained using a labeled training set including pairs of alerts labeled with prioritization results indicating which alert of the pair is higher priority than the other alert of the pair. The prioritization model may optionally be tuned, for example based on inputs provided by a user, in order to account for subsequent developments or otherwise based on additional prioritization considerations which may not be explicitly captured in the prioritization model.

[0029] When the prioritization model is trained, new alerts are ingested and the model is applied to features extracted from the new alerts. Based on the outputs of the model when applied in this manner, an alert priority is determined. In some embodiments, the alert priority may be further determined based on efficiency of remediation actions, for example, by further prioritizing remediation actions which will remediate multiple threats such that the cumulative threat reduction is further improved as compared to the initial priority. The alert priority may be utilized to generate an output for display, to prioritize automated remediation actions, or both.

[0030] To aid in training the prioritization model more efficiently, various disclosed embodiments include techniques for clustering which may be utilized to reduce the amount of alerts to be presented for obtaining prioritization inputs. More specifically, the clustering techniques may be utilized to reduce the sample size of alerts presented to a user in order to obtain prioritization inputs while ensuring that the sampled alerts accurately represent the broader data set. To this end, alerts are clustered based on features distances, and alerts are sampled from each cluster based on proximity to the center of the cluster. In some implementations, clustering as described herein may reduce the number of potential alerts to be used for obtaining prioritization data from millions of alerts to less than 100. Further, groupings of alerts may be selected from different clusters such that each alert in the grouping is selected from a different cluster in order to further improve the efficiency of obtaining prioritization data by avoiding presenting alerts from the same cluster to obtain prioritization inputs.

[0031] In this regard, it is noted that alerts within the same cluster as described herein tend to be prioritized similarly, that is, alerts in the same cluster tend to be given the same priority relative to alerts in other clusters. Accordingly, clustering alerts before obtaining prioritization inputs allows for selecting representative alerts from each cluster and comparing only the representative alerts rather than all of the alerts in each cluster. That is, prioritization may effectively be performed between clusters by comparing only representative alerts between the clusters. Prioritizations between clusters may then be enforced on other alerts within a cluster such that each of the alerts in a first cluster, whose representative alerts were determined to be higher priority than representative alerts of a second cluster, may be assigned higher priority as compared to each alert in the second cluster without needing to compare each alert in the first cluster to each alert in the second cluster.

[0032] Additionally, various disclosed embodiments further provide techniques for determining normalized alert severities. The normalized alert severities provide severity data for alerts in a unified format with standardized severity values, thereby allowing for effectively comparing severities between different data sources. In some embodiments, the normalization may utilize a severity normalization model to determine a normalized severity for each alert. Such a severity normalization model may further be a machine learning model trained based on features extracted from historical alerts. Such features may include, but are not limited to, training alert severities and other training features for aspects of the underlying threats represented by the alerts, and may further include features related to the tools which generated the historical alerts such as frequency of different severity values or user feedback (e.g., user inputs indicating whether a given user agrees or disagrees with the severity).

[0033] In this regard, it is noted that there is no uniform standard for alerting on severity of threats, and that different providers may assign more or less weight to different factors which affect how severe a threat is. As a non-limiting example, the same issue may be alerted by one provider's tool as being "high" severity, and another provider's tool as being "critical" severity. Normalizing the format and values of severity therefore allows for effectively and more accurately prioritizing alerts from such different sources.

[0034] The disclosed embodiments leverage machine learning in order to automate remediation of alerts in a manner which does not require an explicit prioritization procedure from the user. Moreover, the prioritization may be customized to a given user, i.e., to emulate the prioritizations which that user would make. However, the disclosed techniques allow for emulating the user's prioritizations in a more objective manner than decisions which would be taken by the user themselves, and therefore are less influenced by external factors which might affect the user's objectivity.

[0035] The priorities determined in accordance with the disclosed embodiments may therefore be more accurate than naïve priorities based purely on raw incoming severity information indicated in alerts and more consistent than subjective judgments about prioritization which may be performed by manual operators. Accordingly, remedial actions performed based on priorities determined as described herein allow for more effectively securing computing infrastructures against potential cyber threats, thereby improving cybersecurity of systems and data among those computing infrastructures.

[0036] Further, the prioritization model created in accordance with various disclosed embodiments may allow for further explicit customization by the user post training. To this end, some embodiments may utilize a limited feature set for the machine learning model, for example, a number of features less than 10. Such a limited number of features may improve visibility to the user and facilitate user interactions in customizing the model, for example, to customize the model based on updates in prioritization approaches. Further, in some implementations, the prioritization model may avoid using neural network techniques in order to provide a user visibility by avoiding any hidden layers, thereby further facilitating user customization of the model.

[0037] It is also noted that prioritization procedures used by a given user may vary by organization, and may or may not be explicitly documented. Moreover, even if those procedures are documented, they must be translated into a set of rules which explicitly define the decisions used for prioritization (e.g., in the form of an explicitly defined decision tree). Various disclosed embodiments utilize machine learning prioritization models defined with respect to sets of features which can be observed across alerts in order to model user prioritization decisions without requiring explicit programming or otherwise providing of prioritization procedures and processes.

[0038] FIGS. **1**A-B are network diagrams **100**A and **100**B, respectively, utilized to describe various disclosed embodiments.

[0039] FIG. **1**A shows an example network diagram **100**A utilized to describe the various disclosed embodiments. In the example network diagram **100**A, a plurality of cybersecurity tools **120-1** through **120**-N (hereinafter referred to individually as a cybersecurity tool **120** and collectively as cybersecurity tools **120**, merely for simplicity purposes), an alert prioritizer **130**, databases **140**, and an administrator (admin) device **150** communicate via a network **110**. The network **110** may be, but is not limited to, a wireless, cellular or wired network, a local area network (LAN), a wide area network (WAN), a metro area network (MAN), the Internet, the worldwide web (WWW), similar networks, and any combination thereof.

[0040] Each of the cybersecurity tools **120** acts as a source of alerts to be addressed in a computing environment (not shown). To this end, the cybersecurity tools **120** may be configured to perform activities such as, but not limited to, monitoring, collecting data, analyzing, generating alerts, transmitting alerts, and the like. Moreover, in accordance with various disclosed embodiments, the cybersecurity tools **120** may generate alerts in different formats, with different criteria for urgency, or otherwise expressing the same base information in different ways such that the alerts cannot be effectively prioritized via directly comparing data contained therein. As a non-limiting example, alerts indicating a medium urgency from cybersecurity tool **120-1** may, on average, present a higher degree of threat than alerts indicating a high urgency from cybersecurity tool **120-2** such that comparing the urgencies as explicitly indicated in alerts from those cybersecurity tools **120-1** and **120-2** would yield inaccurately prioritizing the alerts.

[0041] The alert prioritizer **130** is configured to prioritize alerts as described herein, including but not limited to, prioritizing alerts from different sources (e.g., different cybersecurity tools **120**). To this end, the alert prioritizer **130** is configured to train and utilize prioritization models trained using training data including alerts (e.g., alerts from the cybersecurity tools **120** or stored in the databases **140**). The training data may further include training prioritization data to be utilized for labeling training inputs for supervised machine learning, for example, labels indicating which alerts are higher priority than others (e.g., labeling each alert in a pair as either higher or lower priority relative to the other alert). Such training prioritization data may be obtained from the admin device **150** or from the databases **140**.

[0042] As discussed herein, the training prioritization data may be obtained using clustering in order to minimize the number of alerts to be utilized for obtaining the training prioritization data. As a non-limiting example, when obtaining the training prioritization data includes presenting pairs of alerts to a user of the admin device **150** and requesting user inputs indicating which alert is higher than the other, the alerts among potential training alerts may be clustered and the alerts to be presented to the user of the admin device **150** are selected from among those potential training alerts based on the clusters. This reduces the amount of interactions needed in order to obtain an effective training set, thereby conserving computing resources as well as reducing the amount of manual work needed in order to obtain a labeled training set for supervised learning.

[0043] The databases **140** may contain, but are not limited to, data to be used for training as described herein. For example, as noted above, the databases **140** may store training alerts, training prioritization data, both, and the like, to be used for training the prioritization model.

[0044] The admin device **150** is operated or otherwise used by an administrator or other user for securing a computing infrastructure (not shown) based on alerts from the cybersecurity tools **120**. To this end, the admin device **150** may be configured to receive and display graphical interfaces illustrating prioritized alerts determined, for example, by the alert prioritizer **130** as described herein.

[0045] In some implementations, the admin device **150** may be utilized for creating training data to be utilized for training the prioritization model as described herein. To this end, in such implementations, the alert prioritizer **130** may send groupings of alerts (e.g., pairs of alerts) to be displayed to a user of the admin device **150**, and the admin device **150** may receive user inputs indicating which alerts are higher priority than other alerts in the same groupings. These user inputs may be utilized to generate training prioritization data to be used for labeling, for example, by labeling each grouping of alerts with the respective user inputs indicating the prioritization or otherwise labeling each grouping of alerts based on the user inputs received in response to displaying the alerts.

[0046] FIG. **1**B shows an example network diagram **1001**B illustrating another deployment in accordance with various disclosed embodiments. More specifically, the network diagram **100**B illustrates an offline deployment in which the alert prioritizer **130** communicates with the databases **140** directly rather than via the Internet or other networks (e.g., the network **110**). Such an offline deployment may allow for training machine learning models using data stored in the databases **140** offline, i.e., without accessing the Internet, which may be relevant for various implementations for further securing data used for training and alert prioritization. To this end, in such an implementation, the databases **140** may contain historical alerts to be used for training.

[0047] FIG. **2** is a flowchart **200** illustrating a method for alert prioritization and remediation according to an embodiment. In an embodiment, the method is performed by the alert prioritizer **130**, FIG. **1**.

[0048] At S**210**, a prioritization model is trained. The prioritization model is a machine learning model trained using training data including training alerts and training prioritization data, where the prioritization data indicates priorities of alerts (e.g., priorities of alerts relative to other alerts among the training alerts). The training alerts are a first set of alerts to be used for training. In an embodiment, the prioritization model is trained as described further below with respect to FIG. **3**.

[0049] As noted above, in some embodiments, the prioritization model may be customized to a particular user or group of users, for example, a particular administrator of a computing environment, a particular business unit, a particular tenant (e.g., users who are employees of an organization), combinations thereof, and the like. In this regard, the prioritization model may be trained using training prioritization data created based on feedback from that user or groupings of user. As a non-limiting example, the training prioritization data may be created based on selections of higher priority alerts among groupings of alerts presented to the user. By using only training prioritization data generated based on inputs made by a given user, the prioritization model may be customized to that user, i.e., such that the prioritization model effectively mimics prioritization decisions which would be made by that user.

[0050] The training prioritization data may be created based on explicit selections made by users or groups of users, based on inferred selections determined based on actions with respect to historical alerts, both, and the like. As another non-limiting example, inferred selections may be determined based on the order in which historical alerts were interacted with in a findings table, which historical alerts were addressed by a user (as opposed to historical alerts which were not ultimately addressed or otherwise skipped), and the like.

[0051] In some embodiments, the prioritization model is trained using further features including actionability features, where the actionability features relate to a degree of actionability for a given alert. Such a degree of actionability may be determined using predetermined actionability determination rules, which may be defined based on root causes of alerts identified via root cause analysis. A non-limiting example root cause analysis process which may be utilized for this purpose is described further in U.S. patent application Ser. No. 17/507,180, assigned to the common assignee, the contents of which are hereby incorporated by reference.

[0052] In a further embodiment, the actionability features are defined with respect to control of the root cause or otherwise location of the root cause. For example, a root cause which is controlled by an organization who will be performing remediation or otherwise located in a computing environment of that organization may be more actionable than a root cause which is controlled or otherwise located outside of the computing environment of that organization (e.g., in a third party's computing environment), and that actionability may be learned based on training actionability features including the controlling entity or location of the root causes for different alerts.

[0053] As a non-limiting example for actionability of an alert related to a vulnerability in a software container, when a root cause of the alert is determined to be in a base image of the software container and the base image is provided by a source outside of a computing environment of an organization which would remediate the alert, then the alert may be less actionable than an alert having a root cause inside of the organization or may not be actionable at all (i.e., since a user within the organization cannot control the base image). Training examples in which an alert had a root cause controlled or located in a certain computing environment may therefore be learned, via machine learning training, to be less actionable than alerts having root causes controlled or located in other computing environments.

[0054] At optional S**220**, the prioritization model may be tuned. The tuning may include, but is not limited to, adjusting weights of the trained prioritization model. As a non-limiting example, the tuning may be performed based on user inputs such as, but not limited to, inputs from an admin viewing the weights of the model. To this end, in some embodiments, the machine learning techniques used to train the prioritization model include using one or more machine learning algorithms which allow for exposing the weights. For example, in an example implementation, use of a neural network (which typically includes nodes with hidden weights) may be avoided in order to ensure that all weights can be exposed, thereby allowing for manual tuning of the trained model.

[0055] At S**230**, alerts are ingested from one or more cybersecurity tools (e.g., the cybersecurity tools **120**, FIGS. **1**A-B). The alerts are a second set of alerts which may indicate or otherwise relate to potential cyber threats in a computing environment such that prioritizing the alerts will allow for more effectively securing the computing environment against cybersecurity threats. More specifically, prioritizing the alerts allows for optimizing use of resources to remediate potential cybersecurity threats, thereby reducing risk in computing environments.

[0056] At S**240**, the trained prioritization model is applied with respect to the alerts. More specifically the prioritization model may be applied to the alerts, or to features extracted from the alerts. To this end, in some embodiments, S**240** further includes extracting such features from the alerts. The prioritization model provides outputs such as, but not limited to, outputs indicating a priority of the alerts.

[0057] At S**250**, an alert priority is determined for the alerts based on the outputs of the prioritization model. The priority may be or may include, but is not limited to, an ordered list of alerts from highest to lowest priority.

[0058] When certain alerts have the same priority, the determined priority may either indicate as such (i.e., that the alerts are the same priority), or predetermined tiebreaking rules may be applied to determine which of the tied alerts (i.e., alerts having the same priority) should be ranked higher in priority. As a non-limiting example, such tiebreaking rules may prioritize alerts related to activity by a particular user who has higher level privileges since potential harm of successful cyber attacks for those alerts would likely cause more damage.

[0059] In some embodiments, the alert priority may be further determined based on efficiency of remediation actions. As a non-limiting example, the alert priority may be determined so as to prioritize remediation actions which will remediate multiple threats such that the cumulative threat reduction is further improved as compared to the initial priority. As a further example, when alerts are within a threshold distance of each other in the ordered list of priority (e.g., alerts are within two places of each other) and it is determined that remediating the lower priority alert will remediate multiple other alerts, then the lower priority alert may be prioritized over the higher priority alert in order to provide an overall more efficient remediation of potential cyber threats within the computing environment.

[0060] As noted above, in a further embodiment, the prioritization model may be further trained based on actionability features such as, but not limited to, features indicating control over or location of a root cause of each alert. In such an embodiment, the prioritization model is further applied to application actionability features for the alerts to be prioritized. To this end, in a further embodiment, S**250** also includes performing root cause analysis in order to determine a root cause of each alert, and determining actionability features (e.g., control, location, etc.) for the root cause of each alert. The prioritization model may then be applied to the determined actionability features. The resulting priority would, in such an embodiment, also account for actionability such that alerts which are more actionable are prioritized over otherwise comparable alerts which are less actionable.

[0061] At S**260**, one or more remediation actions may be performed based on the determined priority. More specifically, remediation actions are performed so as to prioritize remediating higher priority alerts over lower priority alerts. As a non-limiting example, some alerts which are higher priority may be remediated prior to remediating lower priority alerts. As another non-limiting example, only some of the alerts may be remediated, and the determined priority may be utilized to select which alerts to be remediated. As yet another non-limiting example, prioritization suggestions may be generated and presented to a user so that the user may make prioritization decisions based on the suggestions presented based on the determined priority.

[0062] The remediation actions may include, but are not limited to, generating and sending a notification, performing mitigation actions such as changing configurations of software components, changing code of software components, blocking traffic, reconfiguring network components, combinations thereof, and the like. The notification may include, but is not limited to, an ordered list of alerts based on the priority such that the ordered list of alerts may be displayed or otherwise output to a user for use in prioritizing manual remediation actions. The remediation actions to be performed may be determined using one or more predefined remediation rules, which may further be defined with respect to types of issues indicated in the alerts (e.g., issues related to potential cyber threats), locations for which alerts were generated (e.g., locations defined with respect to a computing infrastructure), both, and the like.

[0063] The prioritization of remediation may further be performed based on resources or otherwise based on available capabilities. As a non-limiting example, when remediating an alert includes using remediation actions of a certain cybersecurity tool such that the cybersecurity tool must complete performing remediation actions for one alert before it can proceed to perform remediation actions for another alert, the remediation actions may be performed for respective alerts in order from higher to lower priority. In such embodiments, remediation actions for some lower priority alerts may be performed prior to some higher priority alerts based on availability or capability, that is, if it is possible to remediate the lower priority alerts in parallel with the higher priority alerts but it is not possible to remediate all of the higher priority alerts in parallel, certain lower priority alerts may be remediated earlier than expected according to the ordered list of prioritized alerts. In other words, capabilities for remediation may be leveraged to optimize remediation of as many alerts in parallel as possible regardless of whether some of the lower priority alerts are remediated earlier than they "should" be according to the priority.

[0064] FIG. **3** is a flowchart S**210** illustrating a method for training a prioritization model according to an embodiment.

[0065] At S**310**, the prioritization model is configured. In an embodiment, the configuration is a default configuration which depends on the use case. The configuration may include, but is not limited to, setting initial weights (e.g., to predetermined values), configuring hyperparameters (e.g., based on a predetermined hyperparameter configuration), both, and the like.

[0066] At S**320**, training alerts data is collected. The training alerts data includes training alerts from cybersecurity tools (e.g., the cybersecurity tools **120**, FIGS. **1**A-B). The alerts may indicate, for example, a type of alert (e.g., as defined with respect to an alert identifier), a type of threat which the alert may indicate, a criticality (e.g., a severity such as "high," "low," "medium," "critical," etc.), events which triggered the alert or other data based on which the alert was generated, a time of the alert, combinations thereof, and the like.

[0067] At S**330**, normalized alert severities are determined for the training alerts. As noted above, different cybersecurity tools may express severity in different ways and/or may determined severities using different criteria (i.e., such that the same severity as reported by different tools may not actually represent the same degree of threat). Normalizing the alert severities may help to provide a more accurate reflection of degrees of severity, for example when presenting alerts to users for prioritization when obtaining training prioritization data as discussed below. This, in turn, may reduce bias which might be caused by a user viewing one alert indicating that it is "critical" and another alert indicating that it is "high" when both alerts are actually comparable severity.

[0068] In some embodiments, the normalized alert severities may be determined based on predetermined tendencies known for certain cybersecurity tools (e.g., that "critical" alerts from a particular tool tend to be ultimately considered "high" severity instead), vulnerability-specific context (e.g., by further ranking potential vulnerabilities which may trigger alerts as higher priority than others), both, and the like.

[0069] In an embodiment, S**330** may further include applying one or more machine learning models trained based on historical alerts data to the training alerts in order to determine the normalized alert severities. In an example implementation, such a machine learning model may be trained using supervised machine learning based on a training set including training alerts and training normalized severity labels for respective training alerts. In this regard, such a model may effectively represent learned patterns in overestimating or underestimating severity. As a non-limiting example, when severity alerts from a given cybersecurity tool are consistently overestimated (i.e., the alerts indicate "critical" severity when the normalized severity labels indicate that those alerts were actually only "high" severity), then the normalization machine learning model trained based on that training set may output lower severity normalized alerts for alerts generated by that tool.

[0070] At S**340**, training features are extracted from the training alerts. In an embodiment, the training features include, but are not limited to, types of alerts, severities (e.g., as explicitly indicated by the alerts), both, and the like. The training features may be values representing respective parameters which might relate to alert priorities.

[0071] At S**350**, training prioritization data is obtained for at least a portion of the training alerts. In an embodiment, the training prioritization data is, includes, or is determined based on prioritization inputs provided in response to certain groupings of alerts. As a non-limiting example, alerts may be grouped into pairs, and the training prioritization data may include prioritization inputs indicating which alert in each pair is higher priority than the other. The training prioritization data may be utilized to label the alerts as described herein in order to perform supervised learning.

[0072] It is noted that, in various embodiments, the training prioritization data is not derived solely from analyzing the explicit contents of the alerts. For example, the training prioritization data may be derived from inputs received via a user of a device (e.g., the admin device **150**) in response to presentation of alert groupings. As a result, the prioritization data utilized to train the prioritization model is at least not solely based on the data of the alerts themselves. As noted above, alerts from different cybersecurity tools may indicate severities or other priority-related information using different criteria such that the same or similar severities indicated in alerts from different cybersecurity tools may not accurately reflect the priorities of alerts relative to each other. Thus, obtaining prioritization inputs as described herein and using those prioritization inputs to create the training prioritization data allows for effectively and accurately comparing severities or otherwise prioritizing alerts from different cybersecurity tools, without necessarily requiring explicit data about prioritization of the cybersecurity tools themselves.

[0073] To aid in obtaining the prioritization inputs which allow for prioritizing alerts across different sources (e.g., different cybersecurity tools), obtaining the training prioritization data may further include presenting a reduced set of alerts including samples selected from among a broader data set. To this end, in an embodiment, the training prioritization data for the alerts is obtained at least partially as described further below with respect to FIG. **4**. Specifically, in such embodiments, S**350** may further include clustering the alerts into multiple clusters and selecting one or more representative alerts from each cluster for presentation in order to obtain prioritization inputs to be used for creating the training prioritization data.

[0074] In a further embodiment, representative alerts selected from each cluster are grouped (e.g., into pairs) into groupings containing representative alerts from different clusters such that, when the groupings are presented to a user, the user's selections are effectively selections between clusters, i.e., a user selecting one alert as higher priority over another also demonstrates that other alerts in the cluster of the higher priority alert are likely higher priority than alerts of clusters of lower priority alerts in the same grouping.

[0075] In this regard, it is noted that obtaining the training prioritization data is a technical challenge. That is, providing a smaller data set that is capable of being labeled in a time-effective manner while ensuring that the smaller data set still accurately represents the whole data set poses a technical challenge. The disclosed clustering provides a technique for overcoming this technical challenge in order to efficiently provide accurate training prioritization data. Moreover, such use of clustering reduces computing resources needed to transmit and display larger amounts of training alerts as well as improves user interactions via a streamlined input process.

[0076] In some embodiments, S**350** further includes inferring at least some of the training prioritization data based on historical data related to viewing, remediating, or otherwise responding to alerts. Such inference may be utilized in order to obtain more training data while minimizing user inputs and bias. To this end, the inference may be based on factors such as, but not limited to, time spent viewing alerts, time to respond to alerts, whether remediation actions were performed in response to alerts, combinations thereof, and the like. Such historical data may be, for example but not limited to, log data.

[0077] The inferred training prioritization data may be used to supplement the explicitly provide training prioritization data in order to reduce the amount of data which must be explicitly provided. As a non-limiting example, rules applied in order to make such inferences may be defined so as to automatically infer priorities of alerts meeting certain thresholds in order to determine priorities for alerts which are clearly high priority without requiring presenting those alerts to a user, and alerts which are closer calls (e.g., not meeting one or more of such thresholds) may be presented to the user, thereby only presenting alerts which require finer differentiation for manual labeling.

[0078] As a non-limiting example for inferring priorities, S**350** may include querying data related to viewing of a findings table in order to identify events in which a user viewed the findings table for over 15 seconds, where those events where the user stayed in the findings table over a threshold period of time are determined to be related to higher priority alerts (i.e., because a user viewed findings data for that event for a relatively long period of time). Consequently, alerts which were generated within a certain time period before that event may be inferred to be high priority.

[0079] At optional S**360**, at least some of the training alerts are labeled using data among the training prioritization data. In an embodiment, alerts in each grouping are labeled with relative priorities as compared to other alerts in the grouping, for example as indicated by the training prioritization data.

[0080] As a non-limiting example, when alerts are grouped into pairs, a first alert in the pair is labeled as "higher priority" and a second alert in the pair is labeled as "lower priority." As another non-limiting example, when alerts are grouped into groupings of three, a first alert in the pair is labeled as "highest priority," a second alert in the pair is labeled as "middle priority," and a third alert in the pair is labeled as "lowest priority."

[0081] At S**370**, one or more machine learning algorithms are applied to a training set including at least a portion of the training alerts and the training prioritization data. Such machine learning algorithms may be or may include, but are not limited to, a genetic algorithm or simulated annealing algorithm. The result of applying the machine learning algorithms is a prioritization model trained to prioritize subsequent alerts using those subsequent alerts as inputs.

[0082] FIG. **4** is a flowchart S**350** illustrating a method for obtaining prioritization data for alerts according to an embodiment.

[0083] At S**410**, alerts to be prioritized are identified. In an embodiment, the alerts identified at S**410** include alerts among training alerts data, for example, the data collected as described with respect to S**320**, FIG. **3**.

[0084] At S**420**, a first set of clustering features to be used for clustering the alerts is extracted from the alerts. The clustering features may include, but are not limited to, types of alerts, criticalities of alerts (e.g., as expressed with respect to severities, criticalities, priorities, or other information in the alerts indicating a degree of urgency for each alert), both, and the like. In an embodiment, S**420** may include parsing the alerts and performing natural language process on text in the alerts in order to analyze the alerts for such clustering features.

[0085] At S**430**, the alerts are clustered based on the clustering features. In an embodiment, S**430** includes applying one or more clustering algorithms based on the clustering features. In a further embodiment, the clustering may be based on features distances between alerts, i.e., such that alerts having smaller features distances from each other (e.g., below a predetermined threshold) are clustered into the same cluster.

[0086] To this end, in some embodiments, each alert is represented by a clustering features vector created using the clustering features of that alert, and S**430** may include calculating distances between alerts among the alerts to be prioritized based on their respective clustering features vectors.

[0087] In an embodiment, the clusters are defined such that alerts within each cluster have minimal distances from each other (e.g., below a predetermined threshold) while alerts in different clusters have maximal distance from each other (e.g., above a predetermined threshold). In this regard, it is noted that clusters which are better for training in at least some implementations demonstrate high features similarity within the same cluster while maximizing the distance with sets of features from different clusters. These criteria provide sharper distinctions between clusters when selecting alerts for presentation, thereby allowing for reducing the number of samples to be selected while ensuring that the samples effectively represent the broader data set.

[0088] At S**440**, representative alerts are selected from among the clusters. In an embodiment, the representative alerts include alerts which are closest to a center of each cluster or otherwise which are within a threshold distance of a cluster center.

[0089] To this end, cluster centers may be defined, for example, as a features vector including mean, median, or other values which represent common feature values among alerts the cluster. In such embodiments, distance from the cluster center may be determined by calculating a features distance between the features vector of the cluster center and the features vector of a given alert (e.g., the clustering features vector for the alert as discussed above).

[0090] At S**450**, a second set of grouping features is extracted from among the data including the alerts and, specifically, from data of the representative alerts. The grouping features may include, but are not limited to, features vectors for the alerts (e.g., features vectors of the clustering features as discussed above), a cluster to which each alert belongs to, both, and the like. The grouping features are used to determine groupings to be presented in order to obtain prioritization inputs. For example, the grouping features may be utilized to select groupings of alerts from different clusters.

[0091] At S**460**, the groupings are presented in order to prompt prioritization inputs. In an embodiment, the groupings may be sent to a device (e.g., the admin device **150**, FIGS. **1**A-B) operated or otherwise used by a user or users who will provide example prioritizations for the alerts in the groupings. The device, in turn, may accept user inputs made in response to displaying or otherwise presenting the groupings of the alerts.

[0092] At S**470**, prioritization inputs are obtained to be included in the training prioritization data. As noted above, the prioritization inputs may be received by a device presenting the groupings of alerts such that obtaining the prioritization inputs may be obtained by receiving the inputs from those devices.

[0093] FIG. **5** is an example schematic diagram of an alert prioritizer **130** according to an embodiment. The alert prioritizer **130** includes a processing circuitry **510** coupled to a memory **520**, a storage **530**, and a network interface **540**. In an embodiment, the components of the alert prioritizer **130** may be communicatively connected via a bus **550**.

[0094] The processing circuitry **510** may be realized as one or more hardware logic components and circuits. For example, and without limitation, illustrative types of hardware logic components that can be used include field programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), Application-specific standard products (ASSPs), system-on-a-chip systems (SOCs), graphics processing units (GPUs), tensor processing units (TPUs), general-purpose microprocessors, microcontrollers, digital signal processors (DSPs), and the like, or any other hardware logic components that can perform calculations or other manipulations of information.

[0095] The memory **520** may be volatile (e.g., random access memory, etc.), non-volatile (e.g., read only memory, flash memory, etc.), or a combination thereof.

[0096] In one configuration, software for implementing one or more embodiments disclosed herein may be stored in the storage **530**. In another configuration, the memory **520** is configured to store such software. Software shall be construed broadly to mean any type of instructions, whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise. Instructions may include code (e.g., in source code format, binary code format, executable code format, or any other suitable format of code). The instructions, when executed by the processing circuitry **510**, cause the processing circuitry **510** to perform the various processes described herein.

[0097] The storage **530** may be magnetic storage, optical storage, and the like, and may be realized, for example, as flash memory or other memory technology, compact disk-read only memory (CD-ROM), Digital Versatile Disks (DVDs), or any other medium which can be used to store the desired information.

[0098] The network interface **540** allows the alert prioritizer **130** to communicate with, for example, the cybersecurity tools **120**, the databases **140**, the admin device **150**, and the like.

[0099] It should be understood that the embodiments described herein are not limited to the specific architecture illustrated in FIG. **5**, and other architectures may be equally used without departing from the scope of the disclosed embodiments.

[0100] It is important to note that the embodiments disclosed herein are only examples of the many advantageous uses of the innovative teachings herein. In general, statements made in the specification of the present application do not necessarily limit any of the various claimed embodiments. Moreover, some statements may apply to some inventive features but not to others. In general, unless otherwise indicated, singular elements may be in plural and vice versa with no loss of generality. In the drawings, like numerals refer to like parts through several views.

[0101] The various embodiments disclosed herein can be implemented as hardware, firmware, software, or any combination thereof. Moreover, the software may be implemented as an application program tangibly embodied on a program storage unit or computer readable medium consisting of parts, or of certain devices and/or a combination of devices. The application program may be uploaded to, and executed by, a machine comprising any suitable architecture. Preferably, the machine is implemented on a computer platform having hardware such as one or more central processing units ("CPUs"), a memory, and input/output interfaces. The computer platform may also include an operating system and microinstruction code. The various processes and functions described herein may be either part of the microinstruction code or part of the application program, or any combination thereof, which may be executed by a CPU, whether or not such a computer or processor is explicitly shown. In addition, various other peripheral units may be connected to the computer platform such as an additional data storage unit and a printing unit. Furthermore, a non-transitory computer readable medium is any computer readable medium except for a transitory propagating signal.

[0102] All examples and conditional language recited herein are intended for pedagogical purposes to aid the reader in understanding the principles of the disclosed embodiment and the concepts contributed by the inventor to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the disclosed embodiments, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

[0103] It should be understood that any reference to an element herein using a designation such as "first," "second," and so forth does not generally limit the quantity or order of those elements. Rather, these designations are generally used herein as a convenient method of distinguishing between two or more elements or instances of an element. Thus, a reference to first and second elements does not mean that only two elements may be employed there or that the first element must precede the second element in some manner. Also, unless stated otherwise, a set of elements comprises one or more elements.

[0104] As used herein, the phrase "at least one of" followed by a listing of items means that any of the listed items can be utilized individually, or any combination of two or more of the listed items can be utilized. For example, if a system is described as including "at least one of A, B, and C," the system can include A alone; B alone; C alone; 2A; 2B; 2C; 3A; A and B in combination; B and C in combination; A and C in combination; A, B, and C in combination; 2A and C in combination; A, 3B, and 2C in combination; and the like.

What is claimed is:

1. A method for machine learning training, comprising:

clustering a plurality of cybersecurity alerts into a plurality of clusters based on a features distance between alerts among the plurality of cybersecurity alerts, wherein the plurality of cybersecurity alerts is from a plurality of alert sources;

selecting a subset of the plurality of cybersecurity alerts from each of the plurality of clusters, wherein each cluster has a cluster center, wherein the subset of the plurality of cybersecurity alerts is selected from each of the plurality of clusters based on the cluster center of the cluster;

causing presentation of each subset of the plurality of cybersecurity alerts selected from each of the plurality of clusters via a user device in order to obtain a set of user inputs with respect to each presented subset of the plurality of cybersecurity alerts;

creating a training set based on the user inputs with respect to each presented subset of the plurality of cybersecurity alerts, wherein the training set includes the plurality of cybersecurity alerts and a corresponding plurality of training prioritizations for the plurality of cybersecurity alerts; and

training a prioritization machine learning model to prioritize cybersecurity alerts from the plurality of alert sources using the training set.

2. The method of claim **1**, wherein the plurality of cybersecurity alerts is a plurality of first cybersecurity alerts, further comprising:

applying the prioritization machine learning model with respect to a plurality of second cybersecurity alerts from the plurality of alert sources in order to obtain a set of outputs; and

determining a priority for the plurality of second cybersecurity alerts based on the set of outputs.

3. The method of claim **2**, further comprising:

performing a plurality of remediation actions based on the determined priority.

4. The method of claim **1**, wherein each cybersecurity alert among the subset of each grouping belongs to a different cluster of the plurality of clusters than each other cybersecurity alert among the subset of the grouping.

5. The method of claim **1**, further comprising:

determining a plurality of groupings based on the plurality of clusters, wherein the subset of the plurality of cybersecurity alerts selected from each cluster is determined based on the plurality of groupings.

6. The method of claim **1**, wherein each of the plurality of cybersecurity alerts has a features vector, wherein each cluster center is a features vector determined based on the features vectors of the plurality of cybersecurity alerts.

7. The method of claim **6**, wherein each cluster center includes any of: mean values for a portion of the plurality of cybersecurity alerts, and median values for a portion of the plurality of cybersecurity alerts.

8. The method of claim **1**, wherein each training prioritization is defined relative to at least one other training prioritization of the plurality of training prioritizations from at least one different alert source of the plurality of alert sources.

9. The method of claim **1**, further comprising:

determining a plurality of features vectors for the plurality of cybersecurity alerts; and

determining a plurality of centers of the plurality of clusters, wherein each features distance is between a features vector for one of the plurality of cybersecurity alerts and a features vector for one of the centers of one of the clusters among the plurality of clusters to which the cybersecurity alert belongs.

10. A non-transitory computer readable medium having stored thereon instructions for causing a processing circuitry to execute a process, the process comprising:

clustering a plurality of cybersecurity alerts into a plurality of clusters based on a features distance between alerts among the plurality of cybersecurity alerts, wherein the plurality of cybersecurity alerts is from a plurality of alert sources;

selecting a subset of the plurality of cybersecurity alerts from each of the plurality of clusters, wherein each cluster has a cluster center, wherein the subset of the plurality of cybersecurity alerts is selected from each of the plurality of clusters based on the cluster center of the cluster;

causing presentation of each subset of the plurality of cybersecurity alerts selected from each of the plurality of clusters via a user device in order to obtain a set of user inputs with respect to each presented subset of the plurality of cybersecurity alerts;

creating a training set based on the user inputs with respect to each presented subset of the plurality of cybersecurity alerts, wherein the training set includes the plurality of cybersecurity alerts and a corresponding plurality of training prioritizations for the plurality of cybersecurity alerts; and

training a prioritization machine learning model to prioritize cybersecurity alerts from the plurality of alert sources using the training set.

11. A system for machine learning training, comprising:

a processing circuitry; and

a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:

cluster a plurality of cybersecurity alerts into a plurality of clusters based on a features distance between alerts among the plurality of cybersecurity alerts, wherein the plurality of cybersecurity alerts is from a plurality of alert sources;

select a subset of the plurality of cybersecurity alerts from each of the plurality of clusters, wherein each cluster has a cluster center, wherein the subset of the plurality of cybersecurity alerts is selected from each of the plurality of clusters based on the cluster center of the cluster;

cause presentation of each subset of the plurality of cybersecurity alerts selected from each of the plurality of clusters via a user device in order to obtain a set of user inputs with respect to each presented subset of the plurality of cybersecurity alerts;

create a training set based on the user inputs with respect to each presented subset of the plurality of cybersecurity alerts, wherein the training set includes the plurality of cybersecurity alerts and a corresponding plurality of training prioritizations for the plurality of cybersecurity alerts; and

train a prioritization machine learning model to prioritize cybersecurity alerts from the plurality of alert sources using the training set.

12. The system of claim **11**, wherein the plurality of cybersecurity alerts is a plurality of first cybersecurity alerts, wherein the system is further configured to:

apply the prioritization machine learning model with respect to a plurality of second cybersecurity alerts from the plurality of alert sources in order to obtain a set of outputs; and

determine a priority for the plurality of second cybersecurity alerts based on the set of outputs.

13. The system of claim **12**, wherein the system is further configured to:

perform a plurality of remediation actions based on the determined priority.

14. The system of claim **11**, wherein each cybersecurity alert among the subset of each grouping belongs to a different cluster of the plurality of clusters than each other cybersecurity alert among the subset of the grouping.

15. The system of claim **11**, wherein the system is further configured to:

determine a plurality of groupings based on the plurality of clusters, wherein the subset of the plurality of cybersecurity alerts selected from each cluster is determined based on the plurality of groupings.

16. The system of claim **11**, wherein each of the plurality of cybersecurity alerts has a features vector, wherein each cluster center is a features vector determined based on the features vectors of the plurality of cybersecurity alerts.

17. The system of claim **16**, wherein each cluster center includes any of: mean values for a portion of the plurality of cybersecurity alerts, and median values for a portion of the plurality of cybersecurity alerts.

18. The system of claim **11**, wherein each training prioritization is defined relative to at least one other training prioritization of the plurality of training prioritizations from at least one different alert source of the plurality of alert sources.

19. The system of claim **11**, wherein the system is further configured to:

determine a plurality of features vectors for the plurality of cybersecurity alerts; and

determine a plurality of centers of the plurality of clusters, wherein each features distance is between a features vector for one of the plurality of cybersecurity alerts and a features vector for one of the centers of one of the clusters among the plurality of clusters to which the cybersecurity alert belongs.

* * * * *