GENERATIVE ARTIFICIAL INTELLIGENCE MODEL PROMPT INJECTION CLASSIFIER

RELATED APPLICATIONS

[0001] The current application claims priority to U.S. Pat. App. Ser. No. 18/605,337 filed on Mach 14, 2024 and U.S. Pat. App. Ser. No. 19/038,505, filed on January 27, 2025, the contents of both of which are hereby fully incorporated by reference.

TECHNICAL FIELD

[0002] The subject matter described herein relates to techniques for identifying or otherwise characterizing a prompt injection attack on an artificial intelligence (AI) model, such as a large language model, using a prompt injection classifier.

BACKGROUND

[0003] Machine learning (ML) algorithms and models, such as large language models, ingest large amounts of data and use pattern recognition and other techniques to make predictions and adjustments based on that data. These models have attack surfaces that can be vulnerable to cyberattacks in which adversaries attempt to manipulate or modify model behavior. These attacks can act to corrupt input data so as to make outputs unreliable or incorrect. By modifying or otherwise manipulating the input of a model, an attacker can modify an output of an application or process for malicious purposes including bypassing security measures resulting in data leakage, unauthorized system access, and the solicitation of unlawful or otherwise restricted information.

SUMMARY

[0004] In a first aspect, an analysis engine receives data characterizing a prompt for ingestion by a generative artificial intelligence (GenAI) model. The analysis engine, using a prompt injection classifier, determines whether the prompt comprises malicious

content, elicits malicious actions or other undesired GenAI model behavior. Data characterizing the determination is later provided to a consuming application or process.

[0005] The data characterizing the prompt can be vectorized to result in one or more vectors. One or more embeddings can be generated based on the one or more vectors (which can have a lower dimensionality than the corresponding vectors). The prompt injection classifier can use these embeddings when making the determination. In some variations, the prompt injection classifier uses both features corresponding to the embeddings as well as other features derived from other information.

[0006] The GenAI model can take various forms including a large language model.

[0007] The consuming application or process can take various actions based on the determination. For example, the consuming application or process can allow the prompt to be input into the GenAI model upon a determination that the prompt does not comprise malicious content and/or it can prevent the prompt from being input into the GenAI model upon a determination that the prompt comprises or elicits malicious content. The consuming application or process can flag (e.g., generate an alert, label, etc.) the prompt as being malicious for quality assurance upon a determination that the prompt comprises or elicits malicious content. The consuming application or process can modify the prompt to be benign upon a determination that the prompt comprises or elicits malicious content and can cause the modified prompt to be ingested by the GenAI model. The consuming application or process can block an entity identified by one or more of an internet protocol (IP) address, a media access control (MAC) address, session, or other identifier of a requester of the prompt upon a determination that the prompt comprises or elicits malicious content. The consuming application or process can cause subsequent prompts from the IP address of the requester of the prompt to be modified upon a determination that the prompt comprises or elicits malicious content and can cause the modified prompt to be ingested by the GenAI model. In cases in which an IP address is hosting multiple clients, a combination of the MAC address and the IP address can be blocked or, in other implementations, the HTTP session can be terminated / restarted as well as blocking the MAC address and the IP address.

[0008] In an interrelated aspect, the prompt injection classifier can be a multi-class model. With such an arrangement, an analysis engine receives data characterizing a prompt for ingestion by a generative artificial intelligence (GenAI) model. The analysis engine uses the prompt injection classifier to determine a category for the prompt which is indicative of whether the prompt comprises or elicits malicious content (or otherwise causes the GenAI model to behave in an undesired manner. Data characterizing such a determination can be provided to a consuming application or process.

[0009] The categories can take varying forms. As one example, the category can specify a threat severity for the prompt (malicious, suspicious, unknown, or benign, etc.). In other variations, the category can specify a type of prompt injection attack. The prompt injection attack types can include, for example, one or more of: a direct task deflection attack, a special case attack, a context continuation attack, a context termination attack, a syntactic transformation attack, an encryption attack, a text redirection attack, as well as other types of prompt injection attacks. In these variations, one or more remediation actions can be initiated which are tailored to the specified type of prompt injection attack (as determined by the prompt injection classifier). In some cases, the prompt injection classifier can comprise a plurality of different machine learning models (e.g., an ensemble of machine learning models) in which at least some of such models are trained to categorize a different type of prompt injection attack.

[0010] The consuming application or process can take various actions based on the determined category of the prompt. For example, the consuming application or process can allow the prompt to be input into the GenAI model upon a determination that the prompt is of a category that does not comprise or elicit malicious content. The consuming application or process can prevent the prompt from being input into the GenAI model upon a determination that the prompt is of a category that comprises or elicits malicious content. The consuming application or process can flag the prompt as being malicious for quality assurance upon a determination that the prompt is of a category that comprises or elicits malicious content. The consuming application or process can modify the prompt to be benign upon a determination that the prompt is of a category that comprises or elicits malicious content and can cause the modified prompt to

be ingested by the GenAI model. The consuming application or process can block an IP address of a requester of the prompt upon a determination that the prompt is of a category that comprises or elicits malicious content. The consuming application or process can cause subsequent prompts from an entity identified by an IP address, MAC address, session or other identifier of a requester of the prompt to be modified upon a determination that the prompt is of a category that comprises or elicits malicious content and can cause the modified prompt to be ingested by the GenAI model.

[0011] In a further interrelated aspect, an analysis engine receives data characterizing a prompt for ingestion by a generative artificial intelligence (GenAI) model. The analysis engine, using a prompt injection classifier, determines whether the prompt is indicative of comprising malicious content or eliciting malicious actions. The prompt is allowed to be ingested by the GenAI model if it is determined that the prompt is not indicative of comprising malicious content or eliciting malicious actions. At least one remediation action is initiated if it is determined that the prompt is indicative of comprising malicious content or eliciting malicious actions.

[0012] Non-transitory computer program products (i.e., physically embodied computer program products) are also described that comprise instructions, which when executed by one or more data processors of one or more computing systems, cause at least one data processor to perform operations herein. Similarly, computer systems are also described that may include one or more data processors and memory coupled to the one or more data processors. The memory may temporarily or permanently store instructions that cause at least one processor to perform one or more of the operations described herein. In addition, methods can be implemented by one or more data processors either within a single computing system or distributed among two or more computing systems. Such computing systems can be connected and can exchange data and/or commands or other instructions or the like via one or more connections, including but not limited to a connection over a network (e.g., the Internet, a wireless wide area network, a local area network, a wide area network, a wired network, or the like), via a direct connection between one or more of the multiple computing systems, etc.

[0013] The subject matter described herein provides many technical advantages. For example, the current subject matter can be used to identify and stop adversarial attacks on artificial intelligence models including large language models. Further, the current subject matter can provide enhanced visibility into the health and security of an enterprise's machine learning assets.

[0014] The details of one or more variations of the subject matter described herein are set forth in the accompanying drawings and the description below. Other features and advantages of the subject matter described herein will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0015] FIG. 1 is a system diagram illustrating a security platform for machine learning model architectures having a monitoring environment including an analysis engine and remediation engine;

[0016] FIG. 2 is a system diagram illustrating a security platform for machine learning model architectures having a monitoring environment including an analysis engine which interfaces with external remediation resources;

[0017] FIG. 3 is a system diagram illustrating a security platform for machine learning model architectures having a model environment having a local analysis engine and a monitoring environment including an analysis engine and a remediation engine;

[0018] FIG. 4 is a system diagram illustrating a security platform for machine learning model architectures having a model environment having a local analysis engine and a local remediation engine and a monitoring environment including an analysis engine and a remediation engine;

[0019] FIG. 5 is a system diagram illustrating a security platform for machine learning model architectures having a model environment having a local analysis engine and a local remediation engine and a monitoring environment including an analysis engine;

[0020] FIG. 6 is a system diagram illustrating a security platform for machine learning model architectures having a model environment having a local analysis engine

and a local remediation engine and a monitoring environment including a remediation engine;

[0021] FIG. 7 is a system diagram illustrating a security platform for machine learning model architectures having a model environment having a local analysis engine and a local remediation engine and a monitoring environment including an analysis engine which interfaces with external remediation resources;

[0022] FIG. 8 is a system diagram illustrating a security platform for machine learning model architectures having a model environment having a local analysis engine and a monitoring environment including an analysis engine which interfaces with external remediation resources;

[0023] FIG. 9 is a system diagram illustrating a security platform for machine learning model architectures having a model environment having a local analysis engine and remediation engine;

[0024] FIG. 10 is a system diagram illustrating a security platform for machine learning model architectures having a model environment having a local analysis engine which interfaces with external remediation resources; and

[0025] FIG. 11 is a process flow diagram illustrating monitoring inputs of a machine learning architecture to determine, using a prompt injection classifier, whether the prompt is malicious or elicits malicious actions.

[0026] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0027] The current subject matter is directed to advanced techniques for identifying and preventing cyberattacks on advanced artificial intelligence (AI) models including large language models. In particular, the current subject matter is directed to analyzing prompts of an GenAI model to determine, using machine learning, whether they are malicious or benign, and in some variations, a particular type of prompt injection attack can be identified. Malicious as used herein can refer to actions which cause the GenAI model to respond in an undesired manner. With these classifications, remediation actions

can be taken in connection with the prompt including blocking the prompt, modifying the prompt, disconnecting the requesting device, disconnecting the account, and the like.

[0028] FIG. 1 is a diagram 100 in which each of a plurality of client devices 110 (e.g., an endpoint computing device, a server, etc.) can query, over one or more networks, a machine learning model architecture (MLA) 130 forming part of a model environment 140. These queries can include or otherwise characterize various information including prompts (i.e., alphanumeric strings), videos, audio, images or other files. The model environment 140 can include one or more servers and data stores to execute the MLA 130 and process and respond to queries from the client devices 110. The MLA 130 can comprise or otherwise execute one or more GenAI models utilizing one or more of natural language processing, computer vision, and machine learning. Intermediate the MLA 130 and the client devices 110 is a proxy 150 which can analyze, intercept and/or modify inputs and/or outputs of the MLA 130.

[0029] The proxy 150 can communicate, over one or more networks, with a monitoring environment 160. The monitoring environment 160 can include one or more servers and data stores to execute an analysis engine 170. The analysis engine 170 can execute one or more of the algorithms / models described below with regard to the protection of the MLA 130.

[0030] The proxy 150 can, in some variations, relay received queries to the monitoring environment 160 prior to ingestion by the MLA 130. The proxy 150 can also or alternatively relay information which characterizes the received queries (e.g., excerpts, extracted features, metadata, etc.) to the monitoring environment 160 prior to ingestion by the MLA 130.

[0031] The analysis engine 170 can analyze the relayed queries and/or information in order to make an assessment or other determination as to whether the queries are indicative of being malicious. In some cases, a remediation engine 180 which can form part of the monitoring environment 160 (or be external such as illustrated in FIG. 2) can take one or more remediation actions in response to a determination of a query as being malicious. These remediation actions can take various forms including transmitting data to the proxy 150 which causes the query to be blocked before ingestion by the MLA 130.

In some cases, the remediation engine 180 can cause data to be transmitted to the proxy 150 which causes the query to be modified in order to be non-malicious, to remove sensitive information, and the like. Such queries, after modification, can be ingested by the MLA 130 and the output provided to the requesting client device 110. Alternatively, the output of the MLA 130 (after query modification) can be subject to further analysis by the analysis engine 170.

[0032] The proxy 150 can, in some variations, relay outputs of the MLA to the monitoring environment 160 prior to transmission to the respective client device 110. The proxy 150 can also or alternatively relay information which characterizes the outputs (e.g., excerpts, extracted features, metadata, etc.) to the monitoring environment 160 prior to transmission to the respective client device 110.

[0033] The analysis engine 170 can analyze the relayed outputs and/or information from the MLA 130 in order to make an assessment or other determination as to whether the queries are indicative of being malicious (based on the output alone or based on combination of the input and the output). In some cases, the remediation engine 180 can, similar to the actions when the query analysis above, take one or more remediation actions in response to a determination of a query as being malicious. These remediation actions can take various forms including transmitting data to the proxy 150 which causes the output of the MLA 130 to be blocked prior to transmission to the requesting client device 110. In some cases, the remediation engine 180 can cause data to be transmitted to the proxy 150 which causes the output for transmission to the requesting client device 110 to be modified in order to be non-malicious, to remove sensitive information, and the like.

[0034] FIG. 2 is a diagram 200 in which each of a plurality of client devices 110 (e.g., an endpoint computing device, a server, etc.) can query, over one or more networks, a machine learning model architecture (MLA) 130 forming part of a model environment 140. These queries can include or otherwise characterize various information including prompts (i.e., alphanumeric strings), videos, audio, images or other files. The model environment 140 can include one or more servers and data stores to execute the MLA 130 and process and respond to queries from the client devices 110. The MLA 130 can

comprise or otherwise execute one or more GenAI models utilizing one or more of natural language processing, computer vision, and machine learning. Intermediate the MLA 130 and the client devices 110 is a proxy 150 which can analyze, intercept and/or modify inputs and/or outputs of the MLA 130.

[0035] FIG. 2 is a system diagram 200 illustrating a security platform for machine learning model architectures having a configuration in which the monitoring environment 160 includes an analysis engine 170 which interfaces with external remediation resources 190. In this variation, the monitoring environment 160 does not include a remediation engine 180 but rather communicates, via one or more networks, with external remediation resources 190. The external remediation resources 190 can be computing devices or processes which result in actions such as blocking future requests at the network or user level and/or initiating a remediation action which closes off the impacted system until the malicious action which was output is considered ineffective.

[0036] FIG. 3 is a system diagram 300 illustrating a security platform for machine learning model architectures having a configuration in which the model environment 140 includes a local analysis engine 152 and the monitoring environment 160 includes both an analysis engine 170 and a remediation engine 180. In some cases, one or more of the analysis engine 152 and the remediation engine 180 can be encapsulated or otherwise within the proxy 150. In this arrangement, the local analysis engine 152 can analyze inputs and/or outputs of the MLA 130 in order to determine, for example, whether to pass on such inputs and/or outputs to the monitoring environment 160 for further analysis. For example, the local analysis engine 152 can provide a more computationally efficient local screening of inputs and/or outputs using various techniques as provided herein and optionally, using more lightweight models. If the analysis engine 152 determines that an input or output of the MLA requires further analysis, the input or output (or features characterizing same) are passed to the monitoring environment 160 which can, for example, execute more computationally expensive models (e.g., an ensemble of models, etc.) using the analysis engine 170.

[0037] FIG. 4 is a system diagram 400 illustrating a security platform for machine learning model architectures having a configuration in which the model environment

includes both a local analysis engine 152 and a local remediation engine 154. The monitoring environment 160, in this variation, can include an analysis engine 170 and a remediation engine 180. In this arrangement, the local analysis engine 152 can analyze inputs and/or outputs of the MLA 130 in order to determine, for example, whether to pass on such inputs and/or outputs to local remediation engine 154 to take an affirmative remedial action such as blocking or modifying such inputs or outputs. In some cases, the local analysis engine 152 can make a determination to bypass the local remediation engine 154 and send data characterizing an input or output of the MLA 130 to the monitoring environment 160 for further actions (e.g., analysis and/or remediation, etc.). The local remediation engine 154 can, for example, handle simpler (i.e., less computationally expensive) actions while, in some cases, the remediation engine 180 forming part of the monitoring environment 160 can handle more complex (i.e., more computationally expensive) actions.

[0038] FIG. 5 is a system diagram 500 illustrating a security platform for machine learning model architectures in which the model environment 140 includes a local analysis engine 152 and a local remediation engine 154 and the monitoring environment 160 includes an analysis engine 170 (but does not include a remediation engine). With such an arrangement, any remediation activities occur within or are triggered by the local remediation engine 154 in the model environment 140. These activities can be initiated by the local analysis engine 152 and/or the analysis engine 170 forming part of the monitoring environment. In the latter scenario, a determination by the analysis engine 170 results in data (e.g., instructions, scores, etc.) being sent to the model environment 140 which results in remediation actions.

[0039] FIG. 6 is a system diagram 600 illustrating a security platform 600 for machine learning model architectures in which the model environment 140 includes a local analysis engine 152 and a local remediation engine 154 and the monitoring environment 160 includes a remediation engine 180 (but not an analysis engine). With this arrangement, analysis of inputs or outputs is performed in the model environment by the local analysis engine 152. In some cases, remediation can be initiated or otherwise triggered by the local remediation engine 154 while, in other scenarios, the model

environment 140 sends data (e.g., instructions, scores, etc.) to the monitoring environment 160 so that the remediation engine 180 can initiate one or more remedial actions.

[0040] FIG. 7 is a system diagram 700 illustrating a security platform for machine learning model architectures in which the model environment 140 has a local analysis engine 152 and a local remediation engine 154 while the monitoring environment 160 includes an analysis engine 170 which interfaces with external remediation resources 190. With this arrangement, remediation can be initiated or otherwise triggered by the local remediation engine 154 and/or the external remediation resources 190. With the latter scenario, the monitoring environment 160 can send data (e.g., instructions, scores, etc.) to the external remediation resources 190 which can initiate or trigger the remediation actions.

[0041] FIG. 8 is a system diagram 800 illustrating a security platform for machine learning model architectures in which the model environment 140 includes a local analysis engine 152 and the monitoring environment 160 includes an analysis engine 170 (but does not include a remediation engine). In this arrangement, analysis can be conducted in the monitoring environment 160 and/or the model environment 140 by the respective analysis engines 152, 170 with remediation actions being triggered or initiated by the external remediation resources 190.

[0042] FIG. 9 is a system diagram 900 illustrating a security platform for machine learning model architectures having a model environment 140 has a local analysis engine 152 and a local remediation engine 154. In this arrangement, the analysis and remediation actions are taken wholly within the model environment (as opposed to a cloud-based approach involving the monitoring environment 160 as provided in other variations).

[0043] FIG. 10 is a system diagram illustrating a security platform for machine learning model architectures having a model environment 140 including a local analysis engine 152 which interfaces with external remediation resources 190. In this variation, the analysis of inputs / prompts is conducted local within the model environment 140. Actions requiring remediation are then initiated or otherwise triggered by external

remediation resources 190 (which may be outside of the monitoring environment 160) such as those described above.

[0044] As indicated above, one or more of the analysis engines 152, 170 can include, execute, or otherwise instantiate a prompt injection classifier 192, 194 which, in some variations, is a binary classifier which can identify a prompt as being malicious or benign. In some variations, the prompt injection classifier 192, 194 can be a multi-class classifier which can characterize different aspects of a prompt such as, but not limited to, a level of trustworthiness of the prompt (e.g., malicious, suspicious, benign, etc.). In some variations, the prompt injection classifier 192, 194 can be a multi-class classifier which identifies which of a plurality of different attack types are implicated by an input prompt. Two or more of these prompt injection classifiers 192, 194 can form an ensemble of classifiers (i.e., machine learning models). The ensemble of prompt injection classifiers can be arranged such that two or more of the classifiers are executing in parallel. In other variations, the ensemble of prompt injection classifiers can be arranged such that two or more classifiers are working in sequence. For example, a binary classifier can first analyze a prompt to determine whether the prompt is malicious or benign. If the prompt is classified as being malicious, a multi-class classifier can analyze the prompt to determine a particular type of injection attack. This classification by type can be used to take remediation actions which are specifically tailored to the type of attack. Such an arrangement can also be advantageous when the multi-class classifier is more computationally expensive than the binary classifier (which avoids every prompt being analyzed by the multi-class classifier). Other arrangements can be provided with a lightweight classified being executed by the analysis engine 152 in the model environment 140 and a more computationally expensive model can be executed by the analysis engine 170 in the monitoring environment 160.

[0045] The prompt injection classifier 192, 194 can be a machine learning model such as an XGBoost classification model, a logistic regression model, an XLNet model and the like. In the case of a binary classifier, the prompt injection classifier 192, 194 can be trained using a corpus of data which can include a plurality of benign prompts that do not contain prompt injection information and a plurality of malicious prompts that

contain various character strings (which can include portions of alphanumeric symbols, non-printable characters, symbols, controls, etc.) and the like which encapsulate various sorts of prompt injection. Malicious prompts in this context refer to prompts that cause the prompt injection classifier 192, 194 to exhibit undesired behavior. Benign prompts in this context can refer to prompts that do not cause the prompt injection classifier 192, 194 to exhibit undesired behavior. In some variations, the prompts forming part of the corpus can be labeled with their classification. The model training can be performed by converting the prompts into sentence embeddings which can, amongst other features, be used to train the prompt injection classifier 192, 194.

[0046] In the case of a multi-class classifier, the training corpus for the prompt injection classifier 192, 194 can include different sets of prompts for each category (i.e., severity level, type of attack, etc.) which are labeled with their category (e.g., security level, type of attack, etc.). The prompts can be transformed into sentence embeddings which can be used, amongst other features, to train the prompt injection classifier 192, 194.

[0047] The prompt injection classifier 192, 194 can be periodically retrained as new prompt injection techniques are identified and/or new remediation tools are created. Such an arrangement is advantageous in that the prompt injection classifier 192, 194 can evolve to address the continually changing threat landscape.

[0048] After the prompt injection classifier 192, 194 has been trained, the analysis engine 152, 170 can preprocess incoming prompts so that they are suitable for ingestion by the prompt injection classifier 192, 194. For example, the raw / original prompt is transformed into sentence embeddings and then input into the prompt injection classifier 192, 194 which then results in a model prediction. The model prediction for a binary classifier can predict the confidence of the prompt injection classifier. The output of the model can take varying forms including, for example, a score closer to 1 indicating that the prompt is malicious and a score closer to 0 is indicating that the prompt is benign. The model prediction for the multi-class classifiers can identify a category for the prompt (i.e., a class for which the prompt injection classifier 192, 194 has been trained).

[0049] The multi-class classifier variation of the prompt injection classifier 192, 194 can be used to identify a type of attack and, in some cases, take remedial actions which are specifically tailored to that type of attack (e.g., an attempt to obtain sensitive information or otherwise manipulate an output of the MLA 130). Example attacks include for which the prompt injection classifier 192, 194 can be trained include, but are not limited to: a direct task deflection attack, a special case attack, a context continuation attack, a context termination attack, a syntactic transformation attack, an encryption attack, a text redirection attack and the like. A direct task deflection attack can include, for example, assigning the MLA 130 a persona unrelated to its original purpose and directing it to do something is not intentionally intended to do. A special case attack can include attempts to obfuscate malicious prompts by injecting special case characters randomly or methodically, to confuse the MLA 130 to output a malicious response. A context continuation attack can include providing the MLA 130 with a single prompt or multiple prompts which follow some permutation of a pattern like: benign prompt, malicious prompt, benign prompt, continuation of malicious prompt and which, in combination, can trigger a malicious output. A context termination attack can include provoking a malicious response from the MLA 130 by providing a context and requesting the MLA 130 to essentially "fill in the blanks". A syntactic transformation attack can include manipulation of the syntax or structure of an input to trigger or otherwise stimulate a malicious response. An encryption attack can include encrypting the prompt and tasking the MLA 130 to decrypt the prompt specifying the encryption method. A text redirection attack can include manipulating or redirecting the flow of text-based communications between users or systems. One or more of the model environment remediation engine 154, the monitoring environment remediation engine 180, or the external remediation resources 190 can take or otherwise initiate remediation activities that are specific to the type of attack and/or based on the severity classification for the prompt (e.g., malicious, highly suspicious, unknown, unable to classify, etc.). One remediation activity can be to block the IP address of the requester (i.e., the computing device initiating or otherwise relaying the prompt / input for ingestions by the MLA 130). In some cases, multiple remediation activities can be utilized such as blocking an IP

address in combination with a MAC address or terminating / restarting an HTTP session while also blocking the IP and MAC addresses.

[0050] The IP address can also be used to filter (i.e., modify or otherwise redact) prompts before they are input into the MLA 130. The remediation activities can also include generating alerts (e.g., sysadmin alerts) indicating suspicious / malicious prompts. Further, the remediation activities can include capturing system / process behavior associated with suspicious / malicious prompts for analytics or other tracking purposes.

[0051] FIG. 11 is a diagram 1100 in which, at 1110, data characterizing a prompt or query for ingestion by an AI model, such as a generative artificial intelligence (GenAI) model (e.g., MLA 130, a large language model, etc.) is received. This data can comprise the prompt itself or, in some variations, it can comprise features or other aspects that can be used to analyze the prompt. The received data, in some variations, can be routed from the model environment 140 to the monitoring environment 160 by way of the proxy 150. Thereafter, it can be determined, at 1120, whether the prompt comprises or otherwise attempts to elicit malicious content or actions based on an output of a prompt injection classifier. The prompt injection classifier can be a binary classifier which indicates whether the prompt is malicious or benign. The prompt injection classifier can alternatively be a multi-class classifier which can characterize aspects such as, but not limited to, threat severity level and/or specify the particular type of attack that is being attempted by the prompt. This determination can be performed by the analysis engine 152 and/or the analysis engine 170.

[0052] Data which characterizes the determination can then be provided, at 160, to a consuming application or process. For example, the analysis engine 152 can provide the determination to the remediation engine 154, the analysis engine 170 can provide the determination to the remediation engine 180, the analysis engine 152 can provide the determination to the remediation engine 180, the analysis engine 170 can provide the determination to the external remediation resources 190, the analysis engine 152 can provide the determination to the external remediation resources 190, and/or the determination can be transmitted to or otherwise consumed by a local or remote application or process. The analysis engine 152, 170 in this context can act as a

gatekeeper to the GenAI model by sending information to a consuming application or process which results in preventing prompts deemed to be malicious from being input and allowing prompts deemed to be safe to be input. In some cases, the consuming application or process flags the prompt as being malicious for quality assurance upon a determination that the prompt comprises malicious content. In some cases, it may be desirable to modify a prompt (which can be performed by the consuming application or process) so that it ultimately is non-malicious. For example, only portions of the prompt may be deemed malicious and such aspects can be deleted or modified prior to ingestion by the GenAI model. Other actions can be taken based on the IP address of the requester (such as blocking the prompt, blocking subsequent prompts, modifying subsequent prompts, etc.). Such an arrangement still provides the attacker with an output / response thereby potentially masking the fact that the system identified the response as being malicious.

[0053] Various implementations of the subject matter described herein may be realized in digital electronic circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations may include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor (e.g., CPU, GPU, etc.), which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0054] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and may be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the term "machine-readable medium" refers to any computer program product, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable

signal. The term "machine-readable signal" refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0055] To provide for interaction with a user, the subject matter described herein may be implemented on a computing device having a display device (e.g., a LED or LCD (liquid crystal display) monitor) for displaying information to the user and a keyboard and an input device (e.g., mouse, trackball, touchpad, touchscreen, etc.) by which the user may provide input to the computing device. Other kinds of devices may be used to provide for interaction with a user as well; for example, feedback provided to the user may be any form of sensory feedback (e.g., visual feedback, auditory feedback, or tactile feedback); and input from the user may be received in any form, including acoustic, speech, or tactile input.

[0056] The subject matter described herein may be implemented in a computing system that includes a back-end component (e.g., as a data server), or that includes a middleware component (e.g., an application server), or that includes a front-end component (e.g., a client computer having a graphical user interface or a Web browser through which a user may interact with an implementation of the subject matter described herein), or any combination of such back-end, middleware, or front-end components. The components of the system may be interconnected by any form or medium of digital data communication (e.g., a communication network). Examples of communication networks include a local area network ("LAN"), a wide area network ("WAN"), and the Internet.

[0057] The computing system may include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0058] In the descriptions above and in the claims, phrases such as "at least one of" or "one or more of" may occur followed by a conjunctive list of elements or features. The term "and/or" may also occur in a list of two or more elements or features. Unless otherwise implicitly or explicitly contradicted by the context in which it is used, such a phrase is intended to mean any of the listed elements or features individually or any of

the recited elements or features in combination with any of the other recited elements or features. For example, the phrases "at least one of A and B;" "one or more of A and B;" and "A and/or B" are each intended to mean "A alone, B alone, or A and B together." A similar interpretation is also intended for lists including three or more items. For example, the phrases "at least one of A, B, and C;" "one or more of A, B, and C;" and "A, B, and/or C" are each intended to mean "A alone, B alone, C alone, A and B together, A and C together, B and C together, or A and B and C together." In addition, use of the term "based on," above and in the claims is intended to mean, "based at least in part on," such that an unrecited feature or element is also permissible.

[0059] The subject matter described herein can be embodied in systems, apparatus, methods, and/or articles depending on the desired configuration. The implementations set forth in the foregoing description do not represent all implementations consistent with the subject matter described herein. Instead, they are merely some examples consistent with aspects related to the described subject matter. Although a few variations have been described in detail above, other modifications or additions are possible. In particular, further features and/or variations can be provided in addition to those set forth herein. For example, the implementations described above can be directed to various combinations and subcombinations of the disclosed features and/or combinations and subcombinations of several further features disclosed above. In addition, the logic flows depicted in the accompanying figures and/or described herein do not necessarily require the particular order shown, or sequential order, to achieve desirable results. Other implementations may be within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A computer-implemented method comprising:

receiving, by an analysis engine from a proxy, data characterizing a prompt for ingestion by a generative artificial intelligence (GenAI) model, the analysis engine being executed in a monitoring computing environment and the GenAI model being executed in a model computing environment, the monitoring computing environment being different and remote from the model computing environment;

determining, by the analysis engine using a prompt injection classifier, whether the prompt comprises malicious content or elicits malicious actions, the prompt injection classifier comprising at least one machine learning model trained with a data set of malicious prompts and benign prompts; and

providing data characterizing the determination to a consuming application or process.

2. The method of any claim 1, wherein the prompt injection classifier comprises an ensemble of machine learning-based prompt injection classifiers, a first of the prompt injection classifiers being trained to identify a first type of prompt injection attack and a second of the prompt injection classifiers being trained to identify a second, different type of prompt injection attack.

3. The method of claim 2 further comprising:

initiating, based on the provided data, a remediation action which prevents the prompt from being ingested by the GenAI model to ensure that the GenAI model does not operate in an undesired manner and is based on whether the ensemble of machine learning-based prompt injection classifiers identifies the first type of prompt injection attack or the second type of prompt injection attack.

4. A computer-implemented method comprising:

receiving, by an analysis engine from a model environment executing a generative artificial intelligence (GenAI) model, data characterizing a prompt for ingestion by the GenAI model, the analysis engine executing in a monitoring environment remote from the model environment;

determining, by the analysis engine using an ensemble of machine learning-based prompt injection classifiers, whether the prompt comprises malicious content or elicits malicious actions, a first of the prompt injection classifiers being trained to identify a first type of prompt injection attack and a second of the prompt injection classifiers being trained to identify a second, different type of prompt injection attack; and

providing data characterizing the determination to a consuming application or process to (i) initiate a remediation action to ensure that the GenAI model does not operate in an undesired manner when it is determined that the prompt comprises malicious content or elicits malicious actions and (ii) allow the prompt to be ingested when it is determined that the prompt does not comprise malicious content or elicit malicious actions;

wherein the initiated remediation action prevents the prompt from being ingested by the GenAI model to ensure that the GenAI model does not operate in an undesired manner and is based on whether the ensemble of machine learning-based prompt injection classifiers identifies the first type of prompt injection attack or the second type of prompt injection attack.

5. A computer-implemented method comprising:

receiving, by an analysis engine, data characterizing a prompt for ingestion by a generative artificial intelligence (GenAI) model;

determining, by the analysis engine using a prompt injection classifier, a category for the prompt which is indicative of a type of attack elicited by the prompt, the prompt injection classifier being trained using data sets corresponding to each of a plurality of different categories each corresponding to a different type of attack; and

providing data characterizing the determination to a consuming application or process.

6. The method of claim 5, wherein the category specifies a threat severity for the prompt.

7. The method of claim 6, wherein the threat severity categories comprise one or more of: malicious, suspicious, unknown, or benign.

8. The method of any of claims 5 to 7, wherein the category specifies a type of prompt injection attack.

9. The method of claim 8, wherein the type of prompt injection attack comprises one or more of: a direct task deflection attack, a special case attack, a context continuation attack, a context termination attack, a syntactic transformation attack, an encryption attack, a text redirection attack.

10. The method of any of claims 5 to 9, wherein the prompt injection classifier comprises a plurality of different machine learning models, at least a portion of the different machine learning models being trained to categorize a different type of prompt injection attack.

11. The method of any of claims 5 to 10 further comprising:

initiating one or more remediation actions tailored to the specified type of prompt injection attack.

12. The method of any of the preceding claims further comprising:

vectorizing the received data characterizing the prompt to result in one or more vectors;

generating one or more embeddings based on the one or more vectors, the embeddings having a lower dimensionality than the one or more vectors; and

wherein the prompt injection classifier uses the generated one or more embeddings when making the determination.

13. The method of any of the preceding claims, wherein the GenAI model comprises a large language model.

14. The method of any of the preceding claims, wherein the consuming application or process:

prevents the prompt from being input into the GenAI model upon a determination that the prompt comprises or elicits malicious content; and/or

allows the prompt to be input into the GenAI model upon a determination that the prompt does not comprise or elicit malicious content; and/or

flags the prompt as being malicious for quality assurance upon a determination that the prompt comprises or elicits malicious content; and/or

modifies the prompt to be benign upon a determination that the prompt comprises or elicits malicious content and causes the modified prompt to be ingested by the GenAI model; and/or

blocks an internet protocol (IP) address of a requester of the prompt upon a determination that the prompt comprises or elicits malicious content; and/or

causes subsequent prompts from an entity identified by one or more of an internet protocol (IP) address, a media access control (MAC) address, or a session identifier of a requester of the prompt to be modified upon a determination that the prompt comprises or elicits malicious content and causes the modified prompt to be ingested by the GenAI model.

15. A system comprising:

at least one data processor; and

memory storing instructions which, when executed by the at least one data processor, result in a method as in any of the preceding claims.