

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner

v.

SKYSONG INNOVATIONS, LLC,
Patent Owner

Inter Partes Review Case No. IPR2025-01170
U.S. Patent No. 11,892,897

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 11,892,897**

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	THE '897 PATENT.....	1
	A. The '897 Patent's Alleged Invention	1
	B. The '897 Patent's Prosecution	2
	C. A Person of Ordinary Skill in the Art	3
III.	REQUIREMENTS UNDER 37 C.F.R. § 42.104	4
	A. Standing Under 37 C.F.R. § 42.104(a)	4
	B. Identification of Challenge Under 37 C.F.R. § 42.104(b)	4
	C. Claim Construction Under 37 C.F.R. § 42.104(b)(3)	5
IV.	SHOWING OF ANALOGOUS, PRIOR ART	5
	A. Sabottke.....	5
	B. Smyth	8
	C. Phillips	10
	D. Nunes	11
	E. Kuperman.....	13
	F. Marin.....	14
	G. Bozorgi.....	15
V.	GROUND 1: CLAIMS 1, 3, 5, 9-10, 14-15, 18, 20 ARE OBVIOUS OVER SABOTTKE.....	15
	A. Claim 1	16
	1. Claim 1(Pre): “A method for assessing a likelihood of exploitation of software vulnerabilities, comprising: ”.....	16
	2. Claim 1(a): “utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations, including: ”	17
	3. Claim 1(b): “accessing a plurality of datasets associated with a predetermined set of data sources, the plurality of datasets including training data comprising hacker communications; ”	18
	4. Claim 1(c): “accessing features from the plurality of datasets that include measures computed from social connections of users posting hacking-related content; ”	20

5.	<i>Claim 1(d): “applying learning algorithms to the training data to generate classification models that are configured to predict class labels defining a likelihood of exploitation of respective software vulnerabilities;”</i>	21
6.	<i>Claim 1(e): “accessing one or more features associated with a software vulnerability; and;”</i>	22
7.	<i>Claim 1(f): “computing, by applying the one or more features to the classification model, a class label defining one or more values defining a likelihood of exploitation associated with the software vulnerability”</i>	23
8.	<i>Claim 1(f)(1): “wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.”</i>	24
B.	Claim 3 - The method of claim 1, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.	27
C.	Claim 5 - The method of claim 1, wherein certain features correspond to a known vulnerability obtained from the plurality of datasets.....	28
D.	Claim 9 - The method of claim 1, further comprising computing mutual information from the plurality of datasets informative as to what information a given feature provides about another feature.	28
E.	Claim 10 - The method of claim 1, further comprising: detecting, from the plurality of datasets, vulnerabilities that appear before an associated exploitation date.....	29
F.	Claim 14 - A computing device, configured via machine learning to apply a learned function to data associated with a software vulnerability to estimate a likelihood of exploitation of the software vulnerability, the learned function associated with a prediction model derived from at least one machine learning algorithm and a plurality of datasets associated with software vulnerabilities, the plurality of datasets including information associated with discussion of vulnerabilities by users posting hacking-related content, wherein the likelihood of exploitation predicts an actual exploitation of the software vulnerability before disclosure based on hacker communications from the plurality of datasets.	31

G.	Claim 15 - The computing device of claim 14, wherein the prediction model is at least one classification model that outputs from features of the software vulnerability a score indicative of the likelihood of exploitation.....	32
H.	Claim 18.....	32
1.	<i>Claim 18[Pre]: “A method of prioritizing vulnerabilities using cyber threat intelligence, comprising:”</i>	<i>32</i>
2.	<i>Claim 18(a): “utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations including:”</i>	<i>33</i>
3.	<i>Claim 18(b): “accessing a plurality of datasets associated with a predetermined set of data sources, at least a portion of the plurality of datasets defining training data including hacker communications;”</i>	<i>33</i>
4.	<i>Claim 18(c): “applying learning algorithms to the training data to generate a predictive model configured to predict a likelihood of exploitation of respective software vulnerabilities;”</i>	<i>33</i>
5.	<i>Claim 18(d): “accessing one or more features associated with a software vulnerability; and”</i>	<i>33</i>
6.	<i>Claim 18(e): “computing, by applying the one or more features to the predictive model, one or more values defining a likelihood of exploitation associated with the software vulnerability, wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.”</i>	<i>34</i>
I.	Claim 20 - The method of claim 18, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.....	34
VI.	GROUND 2: CLAIMS 1-3, 5, 10-11, 13-15, 17-20 ARE OBVIOUS OVER SMYTH AND PHILLIPS.....	35
A.	Claim 1.....	35
1.	<i>Claim 1(Pre): “A method for assessing a likelihood of exploitation of software vulnerabilities, comprising:”</i>	<i>35</i>

2.	<i>Claim 1(a): “utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations, including:”</i>	35
3.	<i>Claim 1(b): “accessing a plurality of datasets associated with a predetermined set of data sources, the plurality of datasets including training data comprising hacker communications;”</i>	37
4.	<i>Claim 1(c): “accessing features from the plurality of datasets that include measures computed from social connections of users posting hacking-related content;”</i>	40
5.	<i>Claim 1(d): “applying learning algorithms to the training data to generate classification models that are configured to predict class labels defining a likelihood of exploitation of respective software vulnerabilities;”</i>	44
6.	<i>Claim 1(e): “accessing one or more features associated with a software vulnerability; and;”</i>	47
7.	<i>Claim 1(f): “computing, by applying the one or more features to the classification model, a class label defining one or more values defining a likelihood of exploitation associated with the software vulnerability”</i>	48
8.	<i>Claim 1(f)(1): “wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.”</i>	50
B.	<i>Claim 2 - The method of claim 1, further comprising generating a plurality of estimation outputs based on the one or more values to derive an overall quantitative score.</i>	51
C.	<i>Claim 3 - The method of claim 1, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.</i>	52
D.	<i>Claim 5 - The method of claim 1, wherein certain features correspond to a known vulnerability obtained from the plurality of datasets.</i>	54
E.	<i>Claim 10 - The method of claim 1, further comprising: detecting, from the plurality of datasets, vulnerabilities that appear before an associated exploitation date.</i>	55

F.	Claim 11 - The method of claim 1, further comprising: accessing features from the plurality of datasets that measure a centrality of the users in a social graph.....	55
G.	Claim 13 - The method of claim 1, further comprising vectorizing text features derived from textual content of the plurality of datasets using a predetermined natural language process (NPL).	57
H.	Claim 14 - A computing device, configured via machine learning to apply a learned function to data associated with a software vulnerability to estimate a likelihood of exploitation of the software vulnerability, the learned function associated with a prediction model derived from at least one machine learning algorithm and a plurality of datasets associated with software vulnerabilities, the plurality of datasets including information associated with discussion of vulnerabilities by users posting hacking-related content, wherein the likelihood of exploitation predicts an actual exploitation of the software vulnerability before disclosure based on hacker communications from the plurality of datasets.	57
I.	Claim 15 - The computing device of claim 14, wherein the prediction model is at least one classification model that outputs from features of the software vulnerability a score indicative of the likelihood of exploitation.	59
J.	Claim 17 - The computing device of claim 14, wherein the computing device is further configured to apply the learned function to data associated with each of a plurality of new software vulnerabilities and compute a respective likelihood of exploitation for each of the plurality of new software vulnerabilities.....	61
K.	Claim 18.....	61
1.	<i>Claim 18[Pre]: “A method of prioritizing vulnerabilities using cyber threat intelligence, comprising:”</i>	<i>61</i>
2.	<i>Claim 18(a): “utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations including:”</i>	<i>62</i>
3.	<i>Claim 18(b): “accessing a plurality of datasets associated with a predetermined set of data sources, at least a portion of the plurality of datasets defining training data including hacker communications;”</i>	<i>62</i>

4.	<i>Claim 18(c): “applying learning algorithms to the training data to generate a predictive model configured to predict a likelihood of exploitation of respective software vulnerabilities;”</i>	62
5.	<i>Claim 18(d): “accessing one or more features associated with a software vulnerability; and”</i>	63
6.	<i>Claim 18(e): “computing, by applying the one or more features to the predictive model, one or more values defining a likelihood of exploitation associated with the software vulnerability, wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.”</i>	63
L.	Claim 19 - The method of claim 18, further comprising generating a plurality of estimation outputs based on the one or more values to derive an overall quantitative score.....	63
M.	Claim 20 - The method of claim 18, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.....	64
VII.	GROUND 3: CLAIMS 4 AND 16 ARE OBVIOUS OVER SMYTH, PHILLIPS, AND NUNES.....	64
A.	Claim 4 - The method of claim 3, further comprising: aligning the exploits data with the vulnerability data; and cleaning the exploits data of noise and predetermined portions of the exploits data that is irrelevant to associated software vulnerabilities.....	64
B.	Claim 16 - The computing device of claim 14, wherein the predictive model includes a Random Forest (RF) method including multiple decision tree predictors applied in combination to classify the software vulnerability which is used to estimate the likelihood of exploitation.....	67
VIII.	GROUND 4: CLAIM 6 IS OBVIOUS OVER SMYTH, PHILLIPS, AND KUPERMAN	68
A.	Claim 6 - The method of claim 1, further comprising testing the classification models by applying additional training data and one or more algorithms and evaluation metrics to optimize the	

classification models until the classification models compute
the likelihood of exploitation according to a predefined error
rate.....68

**IX. GROUND 5: CLAIMS 7 AND 13 ARE OBVIOUS OVER
SMYTH, PHILLIPS, AND MARIN.....70**

- A. Claim 7 - The method of claim 1, further comprising
vectorizing text features derived from the plurality of datasets
using term frequency-inverse document frequency to create a
vocabulary of associated words.70
- B. Claim 13 - The method of claim 1, further comprising
vectorizing text features derived from textual content of the
plurality of datasets using a predetermined natural language
process (NPL).72

**X. GROUND 6: CLAIMS 8 AND 12 ARE OBVIOUS OVER
SMYTH, PHILLIPS, AND BOZORGI72**

- A. Claim 8 - The method of claim 1, further comprising: sorting
vulnerabilities associated with the plurality of datasets
according to time; training the classification model using the
training data, the training data defining a first subset of the
plurality of datasets associated with a predetermined period of
time; and testing the classification model using a second subset
of the plurality of datasets associated with the predetermined
period of time.72
- B. Claim 12 - The method of claim 1, further comprising:
accessing one or more features indicative of temporal
connections between at least two of: a time associated with
discussion of a vulnerability by users posting hacking-related
content at a web forum prior to disclosure to a public
vulnerability database; a time associated with disclosure of the
vulnerability to the public vulnerability database; and a time
associated with exploitation of the vulnerability as obtained
through exploits data associated with real-world exploitation of
the vulnerability.74

XI. CONCLUSION.....76

XII. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)77

- A. Real Party-In-Interest.....77
- B. Related Matters77

C.	Lead and Back-Up Counsel	77
D.	37 C.F.R. § 42.8(b)(4) – Service Information.....	78

TABLE OF AUTHORITIES

Cases

<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398, 415-421 (2007).....	67
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303, 1312 (Fed. Cir. 2005).....	5
<i>Skysong Innovations, LLC v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 7:25-cv-00040-DC-DTG	77
<i>Skysong Innovations, LLC v. Fortinet, Inc.</i> , Case No. 2:25-cv-00098.....	77

Statutes

35 U.S.C. § 102(a)(1)	passim
35 U.S.C. § 102(a)(2)	8, 13

Regulations

37 C.F.R. § 42.100(b).....	5
37 C.F.R. § 42.104(a)	4
37 C.F.R. § 42.104(b).....	4
37 C.F.R. § 42.104(b)(3)	5
37 C.F.R. § 42.8(a)(1)	77
37 C.F.R. § 42.8(b)(1)	77, 78
37 C.F.R. §42.24.....	85
37 C.F.R. §42.8.....	85

I. INTRODUCTION

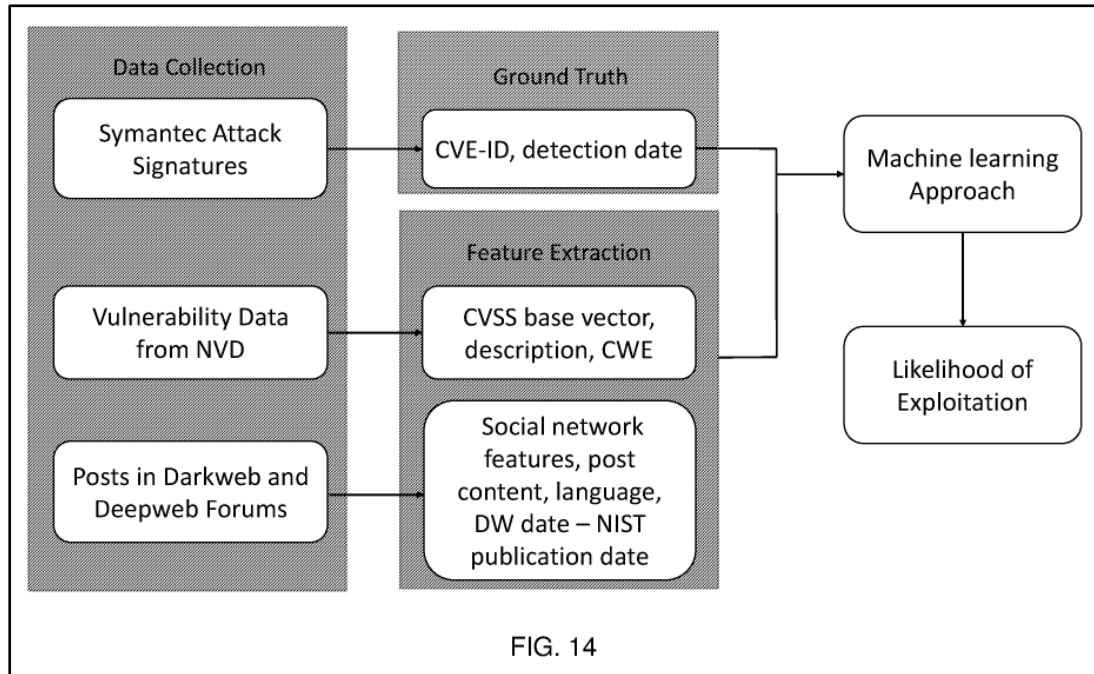
CrowdStrike, Inc. (“Petitioner”) respectfully requests inter partes review (“IPR”) of claims 1-20 (“Challenged Claims”) of U.S. Patent No. 11,892,897 (“’897 Patent”).

II. THE ’897 PATENT

A. The ’897 Patent’s Alleged Invention

The ’897 Patent describes systems and methods “for predicting which software vulnerabilities will be exploited by malicious hackers” to prioritize patching. ’897 *Patent* (Ex. 1001), Abstract, 1:24-28. Methods for prioritizing patching vulnerabilities were admittedly known, thus the ’897 Patent aimed to develop machine learning models “that observe data taken from online discussion posts” “to predict a likelihood of exploitation of a particular vulnerability.” *Id.*, 1:67-2:9; ’897 *File History* (Ex. 1002), at 403. In other words, the invention determines “the likelihood that a given vulnerability mentioned in a data source [e.g., forums, social media posts] will be exploited[.]” *Id.*, at 545.

The invention “leverages machine learning techniques on features derived from the social network of users participating in darkweb/deepweb (DW) forums” to assess the likelihood software vulnerabilities will be exploited. ’897 *Patent*, 4:19-27, claim 1. The patent calls this an “exploit prediction model” shown below in Figure 14:



Id., Fig. 14, 5:66-67. The exploit prediction model collects data from multiple data sources, for example, “crawl[ing] websites on DW, both marketplaces and forums, to collect data relating to malicious hacking.” *Id.*, 7:21-26, 8:25-40. Certain features are extracted from the collected data such as NVD features or social network features. *Id.*, 11:15-38, 21:6-46, Fig. 14. A vulnerability analysis is then performed on these features using machine learning. *Id.*, 14:14-22:41. Finally, the system computes a class label indicating whether a given vulnerability is likely to be exploited in the wild or not. *Id.*

B. The '897 Patent's Prosecution

The '897 Patent claims priority to provisional application 62/581,123, filed November 3, 2017. For purposes of this Petition only, Petitioner applies November 3, 2017, as the priority date for the Challenged Claims.

In response to a first office action, Applicant narrowed the claims to require, in relevant part, accessing from the plurality of datasets “information associated with discussion of vulnerabilities by users posting hacking-related content...prior to disclosure to a public vulnerability database.” ’897 *File History*, 397. The Examiner issued a final rejection, finding accessing hacker forums for vulnerabilities mentioned before disclosure to a public vulnerability database taught by applicant admitted prior art (“AAPA”). *Id.*, 458-459 (“*AAPA discuss vulnerability postings in darkweb forums and ZDI communities from at least 2015...in each case, the...content may be private and submitted before they were disclosed to a public vulnerability database*”). To position the claims for allowance, Applicant filed an RCE adding the following two limitations to claim 1:

accessing features from the plurality of datasets that include measures computed from social connections of users posting hacking-related content

wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.

Id., 478, 517-523 (NOA). As set forth below, these concepts were well-known in the art before the earliest priority date of the ’897 Patent.

C. A Person of Ordinary Skill in the Art

A POSITA at the time of the ’897 Patent would have had a bachelor’s degree in electrical or computer engineering, computer science, or the equivalent, and (a)

either a master's degree in electrical or computer engineering, computer science, or the equivalent, or (b) two years of work/research experience in computer systems and networks. *Dec.*, ¶¶2-14, 37-41.¹ Additional education or experience might substitute for the above requirements. *Id.*

III. REQUIREMENTS UNDER 37 C.F.R. § 42.104

A. Standing Under 37 C.F.R. § 42.104(a)

Petitioner certifies the '897 Patent is eligible for IPR and Petitioner is not barred or estopped from requesting IPR challenging the claims of the '897 Patent. Petitioner is not the owner of the '897 Patent, has not filed a civil action challenging the validity of any claim of the '897 Patent, and this Petition is filed less than one year after Petitioner was served with a complaint alleging infringement of the '897 Patent.

B. Identification of Challenge Under 37 C.F.R. § 42.104(b)

Petitioner requests the Challenged Claims be found unpatentable on the following grounds.

Proposed Grounds of Unpatentability	Exhibits
Ground 1: Claims 1, 3, 5, 9-10, 14-15, 18, 20 are obvious under AIA 35 U.S.C. § 103 over <i>Sabottke</i> .	Ex. 1004
Ground 2: Claims 1-3, 5, 10-11, 13-15, 17-20 are obvious under AIA 35 U.S.C. § 103 over <i>Smyth</i> and <i>Phillips</i> .	Ex. 1005, 1006
Ground 3: Claims 4 and 16 are obvious under AIA 35 U.S.C. § 103 over <i>Smyth</i> , <i>Phillips</i> , and <i>Nunes</i> .	Ex. 1005, 1006, 1007

¹ All citations to "Dec." are to paragraph numbers in Ex. 1003, Declaration of Dr. Nicholas Bambos.

Ground 4: Claim 6 is obvious under AIA 35 U.S.C. § 103 over <i>Smyth, Phillips, and Kuperman</i> .	Ex. 1005, 1006, 1008
Ground 5: Claims 7 and 13 are obvious under AIA 35 U.S.C. § 103 over <i>Smyth, Phillips, and Marin</i> .	Ex. 1005, 1006, 1009
Ground 6: Claims 8 and 12 are obvious under AIA 35 U.S.C. § 103 over <i>Smyth, Phillips, and Bozorgi</i> .	Ex. 1005, 1006, 1010

C. Claim Construction Under 37 C.F.R. § 42.104(b)(3)

In this proceeding, claims are interpreted under the same standard applied by Article III courts (i.e., the *Phillips* standard). 37 C.F.R. § 42.100(b); 83 Fed. Reg. 197 (Oct. 11, 2018); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (*en banc*). Petitioner applies the ordinary and customary meaning of the claim terms as understood by a POSITA.²

IV. SHOWING OF ANALOGOUS, PRIOR ART

A. Sabottke

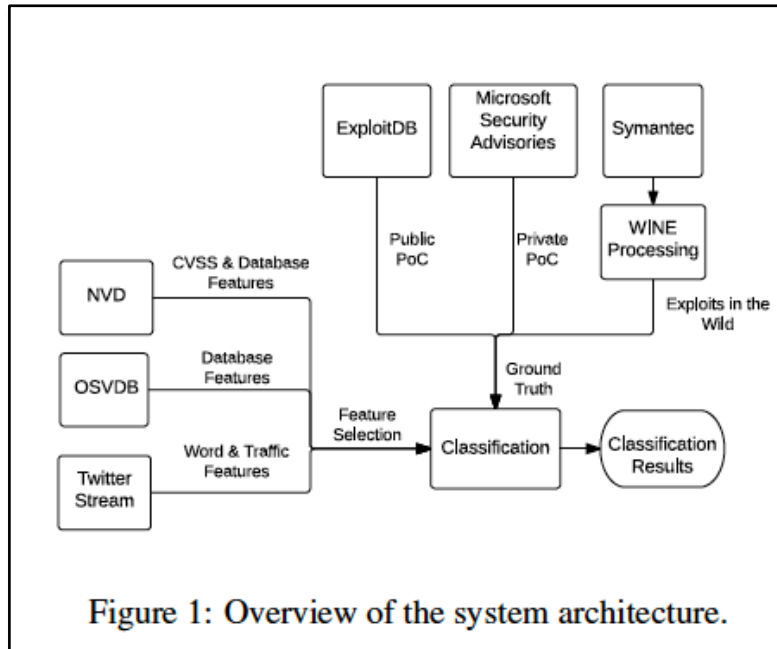
Sabottke published in 24th USENIX Security Symposium and was made available via the USENIX website as of September 23, 2015. *Sabottke* (Ex. 1004); *see also Munford Dec.*, (Ex. 1014), ¶¶49-51. *Sabottke* is prior art under at least 35 U.S.C. § 102(a)(1).

Sabottke teaches a “Twitter-based exploit detector” that “assess[es] which vulnerabilities are likely to be exploited” to prioritize a response. *Sabottke*, 2.

² Petitioner’s challenge is not a waiver of any argument in litigation that any claim term in the ’897 Patent is indefinite or invalid. Petitioner does not waive its right to seek to construe claim terms in a different proceeding.

“[H]ackers...discuss vulnerabilities on social media sites like Twitter” and thus these sites “constitute rich sources of information” for exploit prediction. *Id.* To effectuate “early exploit detection using information available on Twitter,” *Sabottke* utilizes “supervised machine learning and ground truth about exploits.” *Id.* *Sabottke* collects a corpus of tweets and queries the NVD and OSVDB databases for context about the vulnerabilities discussed on Twitter, “extract[s] features for training and testing a support vector machine (SVM) classifier,” and uses the SVM for exploit prediction. *Id.*, 3, 5, 9.

Sabottke provides “early detection of such exploits by using information that is available publicly [e.g., from hacker Twitter streams], but is not included in existing vulnerability databases.” *Id.*, 3. In this way, *Sabottke* can predict vulnerabilities likely to be exploited prior to their disclosure to public vulnerability databases. *Id.*, 9 (“Information posted before disclosure”). Figure 1 below provides an overview of *Sabottke*’s system architecture:



Id., Fig. 1.

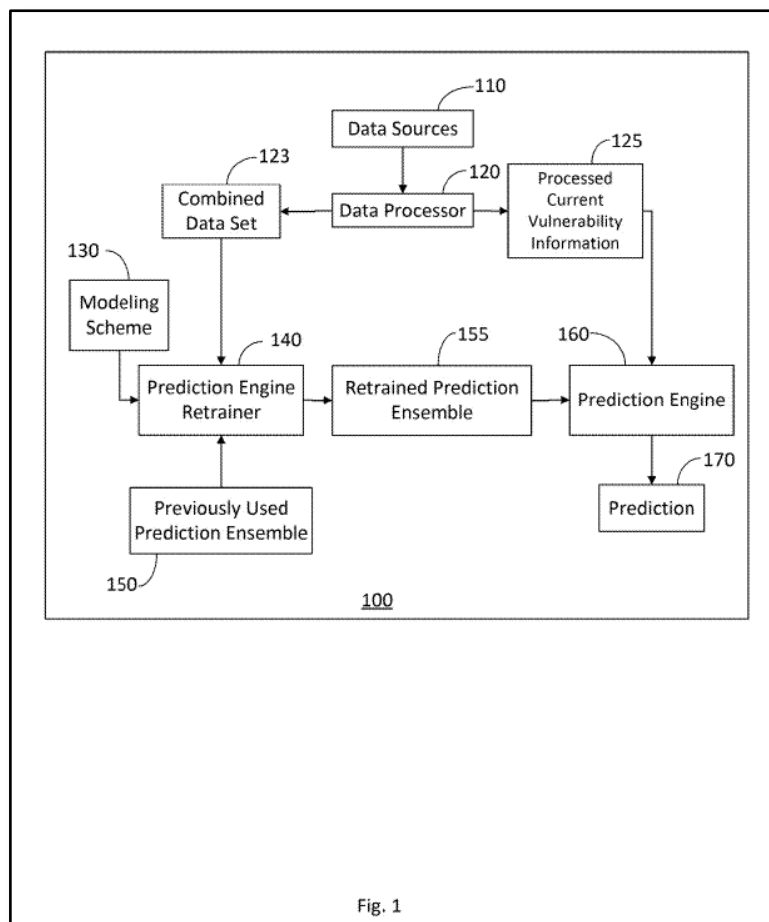
Sabottke, like the '897 Patent, discloses a system for assessing cybersecurity risks of vulnerabilities. *Compare Sabottke*, Abstract, 2-3, 5 with '897 Patent, 1:24-28, 4:1-27, 4:41-44. *Sabottke* is in the same field of endeavor as the '897 Patent. *Dec.*, ¶¶51-53.

Sabottke is also reasonably pertinent to certain problems addressed by the '897 Patent, such as using online discussions to predict whether certain vulnerabilities will be exploited. *Compare Sabottke*, 2-3, 5, 6-8, 16-17 with '897 Patent, 4:19-27, 8:41-45, 13:47-50. *Sabottke* is therefore analogous art to the '897 Patent. *Dec.*, ¶¶51-54.

B. Smyth

U.S. Patent No. 10,264,009 to Smyth et al. (“*Smyth*”) was filed July 26, 2016, qualifying as prior art to the ’897 Patent under at least 35 U.S.C. § 102(a)(2). *Smyth* (Ex. 1005).

Smyth teaches a “predictive engine for analyzing existing vulnerability information to determine the likelihood of a vulnerability being exploited by malicious actors[.]” *Smyth*, 1:7-11. An embodiment using this predictive engine is depicted in Figure 1 below:



Id., Fig. 1. Data processor 120 collects information from a “plurality of data sources 110[.]” *Id.*, 5:40-52. Data sources 110 can include data from the U.S. National Vulnerability Database (NVD), the common vulnerabilities and exposures (CVE) system, and other public or proprietary data sources such as “online discussions” and “discussion forums.” *Id.*, 3:35-6:16, Fig. 2 (listing example data sources). The system then trains predictive models with this data, the models are combined into a “prediction ensemble 155 for use by the prediction engine 160[.]” and a “prediction 170” is generated based on the inputs that indicates “the likelihood the current vulnerability” will be exploited. *Id.*, 4:4-34. Namely, “prediction engine 160 generates a prediction based on processed current vulnerability information 125,” such as online discussion data, using “prediction ensemble 155.” *Id.*, 10:61-64, 11:3-8 (“current vulnerability information 125 may comprise...more current information such as online discussions 240”), 14:33-15:14.

Smyth, like the '897 Patent, discloses a system for assessing cybersecurity risks of vulnerabilities. *Compare Smyth*, Abstract, 1:7-11, 2:25-50, 4:4-6 with '897 Patent, 1:24-28, 4:1-27, 4:41-43. *Smyth* is in the same field of endeavor as the '897 Patent. *Dec.*, ¶¶88-89.

Smyth is also reasonably pertinent to certain problems addressed by the '897 Patent, such as using online forum discussions to predict whether certain vulnerabilities will be exploited. *Compare Smyth*, 2:6-27, 4:44-47, 5:10-15 with '897

Patent, 4:19-27, 8:41-45, 13:47-50. *Smyth* is therefore analogous art to the '897 Patent. *Dec.*, ¶¶88-90.

C. Phillips

Phillips published in SOTICS 2015: The Fifth International Conference on Social Media Technologies, Communication, and Informatics and was available to the public at least by August 3, 2016. *Phillips* (Ex. 1006); *see also Munford Dec.*, (Ex. 1014), ¶¶5-18. *Phillips* was also published on the Oxford University Research Archive and made available to the public on that website at least by December 21, 2015. *Id.* *Phillips* is prior art under at least 35 U.S.C. § 102(a)(1).

Phillips teaches collecting “metadata” associated with “Dark Web Forums” to assess “the social structure of the networks.” *Phillips*, 1. *Phillips* applies various “Social Network Analysis (SNA) techniques in order to identify a range of potentially ‘important’ members of the forums.” *Id.* In these techniques, forum posts are extracted and metadata is pulled from each forum to build a network; a social network is created representing each forum; SNA metrics are run on these social networks; and unsupervised machine learning is applied to identify important forum members. *Id.*, 2. An outline of this approach is shown below:



Id. Example SNA metrics ran on this metadata are described in Table I. *Id.*, 3. These include metrics described in the '897 Patent's specification. *Id.*, (describing, e.g., "Degree Centrality[,] "Betweenness Centrality Score[,] "Pagerank Score"), '897 Patent, 21:6-46 ("[i]n-degree" and "[o]ut-degree centrality[,] "[b]etweenness centrality[,] "[p]agerank centrality").

Phillips is reasonably pertinent to certain problems addressed by the '897 Patent, such as analyzing social connections of users posting hacker-related content in forums to identify security risks. *Compare Phillips*, 1-2 (describing applying "SNA metrics" to metadata from dark web forums), 3 (Table I) with '897 Patent, 4:19-27, 11:15-38, 21:6-46; '897 File History, 404. *Phillips* is therefore analogous art to the '897 Patent. *Dec.*, ¶¶91-92.

D. Nunes

Nunes published in Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics and was available to the public at least by November 8,

2016. *Nunes* (Ex. 1007); *see also* *Munford Dec.*, ¶¶31-34. *Nunes* is prior art under at least 35 U.S.C. § 102(a)(1).

Nunes teaches “an operational system for cyber threat intelligence gathering from various social platforms on the Internet, particularly sites on the darknet[.]” *Nunes*, Abstract. *Nunes*’ system includes a “parser” that “extract[s] specific information from...hacker forums (discussion regarding services and threats).” *Id.*, 2. *Nunes* filters noise from the extracted information using classifiers “to filter out products and topics relating to drugs, weapons, etc. not relevant to malicious hacking.” *Id.*, 3.

Nunes, like the ’897 Patent, discloses a system for assessing cybersecurity risks of vulnerabilities. *Compare Nunes*, Abstract, 1 (determining “[w]hat vulnerabilities do the latest exploits target”), 5, *with* ’897 Patent, 4:19-27, 8:41-45, 13:47-50. *Nunes* is in the same field of endeavor as the ’897 Patent. *Dec.*, ¶¶144-145.

Nunes is also reasonably pertinent to certain problems addressed by the ’897 Patent, such as using online forum discussions to predict whether certain vulnerabilities will be exploited. *Compare Nunes*, Abstract, 1, *with* ’897 Patent, 4:19-27, 8:41-45, 13:47-50. *Nunes* is therefore analogous art to the ’897 Patent. *Dec.*, ¶¶144-146.

E. Kuperman

U.S. Patent No. 10,652,254 to Kuperman et al. (“*Kuperman*”) was filed on Feb. 23, 2017 and claims priority to an application filed on February 23, 2016, qualifying as prior art to the ’897 Patent under at least 35 U.S.C. § 102(a)(2). *Kuperman* (Ex. 1008).

Kuperman teaches systems for detecting malicious activity and protecting web applications. *Kuperman*, Abstract, 1:18-21 3:30-42. *Kuperman* includes a model that “reduces the likelihood of successful exploit 113 of web application 120 vulnerability 121 due to false negatives and reduces the likelihood of false positives 115.” *Id.*, 13:47-52. To this end, *Kuperman* teaches an optimization process for this model that evaluates a desired performance level based on accuracy metrics including “False Positive %” and “False Negative %” such that a test and training error rate determines whether the model exceeds a predefined “desired performance level” and can be marked “live” for use. *Kuperman*, 19:61-20:22, Fig. 6.

Kuperman, like the ’897 Patent, discloses a system for assessing cybersecurity risks for vulnerabilities. Compare *Kuperman*, Abstract, 13:47-52, Fig. 1 with ’897 Patent, 1:24-28, 4:1-27, 4:41-44. *Kuperman* is in the same field of endeavor as the ’897 Patent. *Dec.*, ¶¶153-154.

Kuperman is also reasonably pertinent to certain problems addressed by the ’897 Patent, such as optimizing classification models by increasing the accuracy and

decreasing the false positive identification of exploits. *Compare Kuperman*, 19:61-20:22, Fig. 6 with '897 Patent, 3:54-58, 14:42-15:10, 20:8-11. *Kuperman* is therefore analogous art to the '897 Patent. *Dec.*, ¶¶153-155.

F. Marin

Marin published in Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics and was available to the public at least by November 8, 2016. *Marin* (Ex. 1009); see also *Munford Dec.*, ¶¶27-30. *Marin* is prior art under at least 35 U.S.C. § 102(a)(1).

Marin teaches analyzing websites on the darkweb for malicious hacking products and using machine learning to categorize and study those products. *Marin*, 1. *Marin*'s system includes clustering and grouping categories of online hacking tools, including "exploit kits[.]" *Marin*, 3.

Marin is reasonably pertinent to certain problems addressed by the '897 Patent, such as identifying and classifying malicious activity on the darkweb to better understand active hacker threats. *Compare Marin*, Abstract, 3 (determining prevalence of certain hacking techniques, such as "keyloggers") with '897 Patent, 6:50-59, 7:21-26, 8:25-9:17. *Marin* is therefore analogous art to the '897 Patent. *Dec.*, ¶¶161-162.

G. Bozorgi

Bozorgi published in Proceedings of the 2010 ACM SIGKDD International Conference on Knowledge Discovery and was made available to the public at least by February 28, 2014. *Bozorgi* (Ex. 1010); *see also Munford Dec.*, ¶¶23-26. *Bozorgi* is prior art under at least 35 U.S.C. § 102(a)(1).

Bozorgi teaches a vulnerability prioritization scheme that predicts “whether and how soon a vulnerability is likely to be exploited.” *Bozorgi*, 1-2. *Bozorgi* further teaches extracting various vulnerability features for classifying and prioritizing the vulnerabilities. *Id.*, 4.

Bozorgi, like the '897 Patent, discloses a system for assessing cybersecurity risks of vulnerabilities. *Compare Bozorgi*, Abstract, 2, 5-6, *with '897 Patent*, 1:24-28, 4:1-27, 4:41-44. *Bozorgi* is in the same field of endeavor as the '897 Patent. *Dec.*, ¶¶167-168.

Bozorgi is also reasonably pertinent to certain problems addressed by the '897 Patent, such as prioritizing vulnerabilities for purposes of predicting exploitation. *Compare Bozorgi*, Abstract, 6-8, *with '897 Patent*, 7:13-17, Fig. 13. *Bozorgi* is therefore analogous art to the '897 Patent. *Dec.*, ¶¶167-169.

V. GROUND 1: CLAIMS 1, 3, 5, 9-10, 14-15, 18, 20 ARE OBVIOUS OVER SABOTTKE

The '897 Patent attempts to distinguish prior art methods that rely on social media for exploit prediction. '897 Patent, 1:67-2:10. Namely, the '897 Patent

highlights the benefits of mining hacker posts from the dark web, alleging dark web posts provide more useful and predictive information than public facing social media posts. *Id.*, 3:48-53, 8:25-45, 9:35-37, 11:10-12; *see also id.*, 15:35-46 (comparing dark web embodiment with prior social media approaches because “social media currently is the best” approach). The ’897 Patent’s purported contribution to the art is “predicting exploits in the wild using DW data.” *Id.* Critically, this dark web focus was never reflected in the claims. The claims broadly recite accessing and analyzing hacker communications but fail to narrow those communications to posts on the dark web. Accordingly, the claims read on the prior art systems that analyze hacker communications in social media posts.

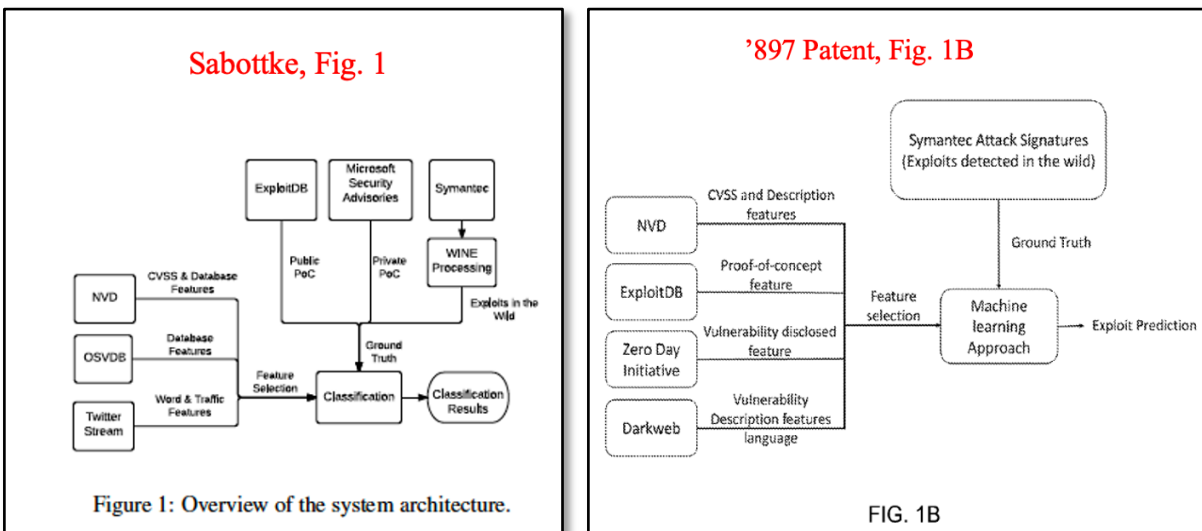
Petitioner proposes a rejection based on *Sabottke*, a reference that expressly teaches the prior art social media methods the claimed invention failed to distinguish.

A. Claim 1

1. Claim 1(Pre): “A method for assessing a likelihood of exploitation of software vulnerabilities, comprising:”

To the extent the preamble is limiting, *Sabottke* teaches a “Twitter-based exploit detector” that assesses the likelihood software vulnerabilities will be exploited. *Sabottke*, 2. *Sabottke* notes the growing number of “**software vulnerabilities**” has created a “need for prioritizing the response to new disclosures by assessing which vulnerabilities are **likely to be exploited**” and proposes a

“Twitter-based exploit detector” to fill that need. *Id.*; *see also id.*, 13 (“Twitter data provides valuable insights into **the likelihood of exploitation**”), 15 (“**likelihood of real-world exploitation**, produced by our system”). *Sabottke*’s exploit prediction system architecture is substantially similar to the ’897 Patent’s exploit prediction model:



Sabottke, Fig. 1; ’897 Patent, Fig. 1B; *Dec.*, ¶¶55-57.

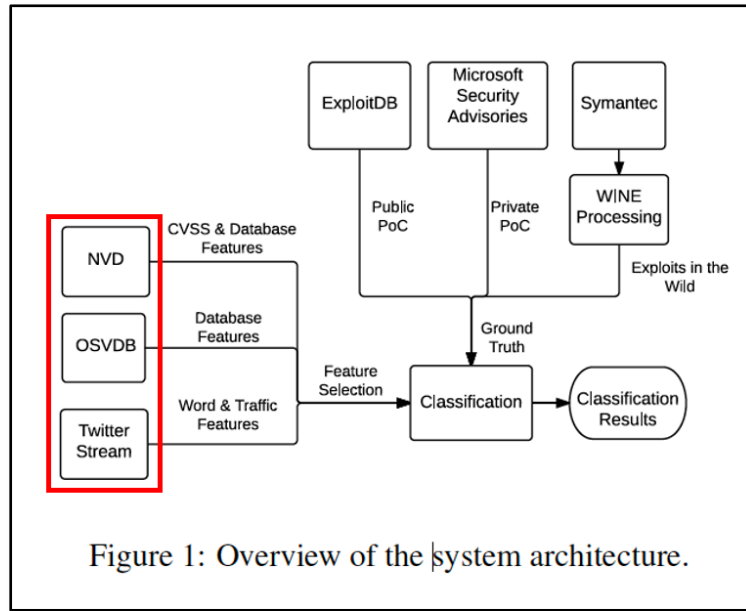
2. ***Claim 1(a): “utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations, including:”***

Sabottke’s “Twitter-based exploit detector” is a software model that utilizes “Twitter’s Streaming API[,],” Python scripts, machine learning, and other software features for its exploit detection methods. *Sabottke*, 5 (“We present the design of a Twitter-based exploit detector, using supervised machine learning techniques” and describing Twitter API and “Python scripts” for data collection). A POSITA would

have understood *Sabottke* suggests the hardware to execute the software instructions associated with its system. *Dec.*, ¶58. Namely, a POSITA would have understood *Sabottke*'s "Twitter-based exploit detector" would have been executable on a general computing system available before 2017, including distributed computing systems, comprising a processor executing software instructions stored in memory for effectuating *Sabottke*'s disclosures. *Dec.*, ¶¶43-44, 59. At a minimum, a POSITA would have found it obvious to implement *Sabottke*'s "Twitter-based exploit detector" on a known computing system by executing software instructions in memory via a CPU. *Dec.*, ¶60.

3. *Claim 1(b): "accessing a plurality of datasets associated with a predetermined set of data sources, the plurality of datasets including training data comprising hacker communications;"*

Sabottke teaches accessing a plurality of datasets from predetermined data sources, such as Twitter feeds and the NVD and OSVDB databases. These datasets are shown below in Figure 1:



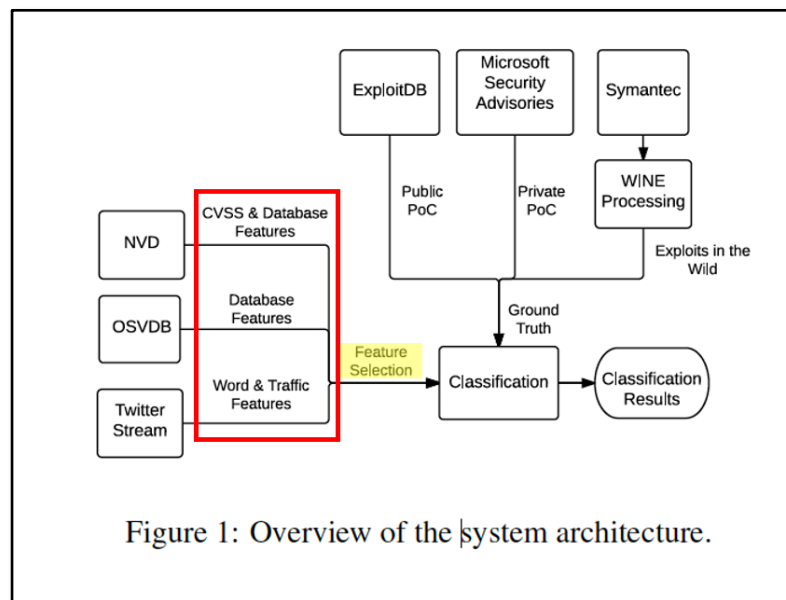
Sabottke, Fig. 1 (annotated). *Sabottke*’s “detector extracts vulnerability-related information from the Twitter stream, and augments it with additional sources of data about vulnerabilities and exploits.” *Id.*, 5. *Sabottke* “collect[s] tweets mentioning vulnerabilities” using “Twitter’s Streaming API” and Python scripts. *Id.* *Sabottke* augments with additional sources to “obtain context about the vulnerabilities discussed on Twitter,” including querying “the National Vulnerability Database (NVD)” for CVSS scores, products affected, and additional references about these vulnerabilities, and crawling the “Open Sourced Vulnerability Database (OSVDB)” for “disclosure dates and categories of vulnerabilities[.]” *Id.*

From the corpus of collected tweets, *Sabottke* “extract[s] features for training and testing a support vector machine (SVM) classifier.” *Id.*, 2-3. *Sabottke* expressly teaches collecting tweets on Twitter that include hacker communications relevant to

security vulnerabilities. *Id.*, 2 (“The security vendors, system administrators, and **hackers**, who discuss vulnerabilities on social media sites like Twitter, constitute rich sources of information” for *Sabottke*’s system), 3 (“Specifically, we analyze the Twitter stream...On Twitter, a community of **hackers**, security vendors and system administrators discuss security vulnerabilities”).

4. ***Claim 1(c): “accessing features from the plurality of datasets that include measures computed from social connections of users posting hacking-related content;”***

Sabottke teaches accessing features from the plurality of datasets during feature selection. *Sabottke*, 2-3 (“we extract features for training and testing a support vector machine (SVM) classifier). *Sabottke*’s feature selection process is depicted below:



Id., Fig. 1. *Sabottke* accesses four categories of features: “Twitter Text, Twitter Statistics, CVSS Information and Database Information.” *Id.*, 6. The specific features are listed below Figure 2:

Figure 2: Mutual information between real world and public proof of concept exploits for CVSS, Twitter user statistics, NVD, and OSVDB features. **CVSS Information** - 0: CVSS Score, 1: Access Complexity, 2: Access Vector, 3: Authentication, 4: Availability Impact, 5: Confidentiality Impact, 6: Integrity Impact. **Twitter Traffic** - 7: Number of tweets, 8/9: # users with minimum T followers/friends, 10/11: # retweets/replies, 12: # tweets favorited, 13/14/15: Avg # hashtags/URLs/user mentions per tweet, 16: # verified accounts, 17: Avg age of accounts, 18: Avg # of tweets per account. **Database Information** - 19: # references in NVD, 20: # sources in NVD, 21/22: BUGTRAQ/SECUNIA in NVD sources, 23: allow in NVD summary, 24: NVD last modified date - NVD published date, 25: NVD last modified date - OSVDB disclosed date, 26: Number of tokens in OSVDB title, 27: Current date - NVD last modified date, 28: OSVDB in NVD sources, 29: code in NVD summary, 30: # OSVDB entries, 31: OSVDB Category, 32: Regex Category, 33: First vendor in NVD, 34: # vendors in NVD, 35: # affected products in NVD.

Id., 8. *Sabottke*’s “Twitter Traffic” features include measures computed from social connections of users posting hacking-related content. Namely, at least the “# users with minimum T followers/friends,” “# retweets/replies,” and “# tweets favorited” features represent measures computed from the poster’s social connections. *Dec.*, ¶63.

5. ***Claim 1(d): “applying learning algorithms to the training data to generate classification models that are configured to predict class labels defining a likelihood of exploitation of respective software vulnerabilities;”***

Sabottke teaches applying supervised learning algorithms to train its exploit detector such that the exploit detector predicts class labels defining a likelihood of exploitation of respective software vulnerabilities. *Sabottke*, 2, 4, 7. Specifically, *Sabottke*’s “techniques utilize supervised machine learning” to train linear support vector machine classifiers. *Id.*; see also *id.*, 5 (“We present the design of a Twitter-

based exploit detector, using supervised machine learning techniques”), 7 (“We train linear support vector machine (SVM) classifiers [] in a feature space” “using multiple rounds of stratified random sampling”). *Sabottke’s* supervised training techniques for its SVM classifiers are “learning algorithms” that generate a trained SVM (i.e., *classification model*). *Dec.*, ¶64.

Sabottke’s trained SVM classifiers are configured to predict class labels defining a likelihood of exploitation of respective software vulnerabilities. *Sabottke* “aim[s] to train a classifier that produces binary predictions: each vulnerability is classified as either exploited or not exploited.” *Id.*, 4. This teaching is consistent with the disclosures of the ’897 Patent. The ’897 Patent provides exemplary classification models, including “Support Vector Machine[,]” that apply a “class label[,]” which is a label that conveys a value of “exploited or not exploited[.]” ’897 *Patent*, 14:30-33, 17:64-65, 21:57-59, 22:66-23:32. Analogously, *Sabottke’s* trained SVMs are a classification model configured to predict class labels as claimed. *Dec.*, ¶65.

6. Claim 1(e): “accessing one or more features associated with a software vulnerability; and;”

Sabottke teaches accessing features associated with software vulnerabilities, including CVSS information and vulnerability database information from NVD and OSVDB:

To obtain context about the vulnerabilities discussed on Twitter, we **query the National Vulnerability Database (NVD) [7] for the CVSS scores, the products affected and additional references**

about these vulnerabilities. Additionally, we crawl the Open Sourced Vulnerability Database (OSVDB) [9] for a few additional attributes, including the disclosure dates and categories of the vulnerabilities in our study.

Sabottke, 5. Regarding CVSS features, *Sabottke* accesses “CVSS Information - 0: CVSS Score, 1: Access Complexity...” *Id.*, 8. For vulnerability database features, *Sabottke* accesses “Database Information - 19: # references in NVD, 20: # sources in NVD...” *Id.*

7. ***Claim 1(f): “computing, by applying the one or more features to the classification model, a class label defining one or more values defining a likelihood of exploitation associated with the software vulnerability”***

Sabottke teaches or suggests computing, by applying the one or more features to the SVM classification model, a class label (e.g., “exploitability”) defining one or more values (e.g., “exploited,” “not exploited”) defining a likelihood of exploitation associated with the software vulnerability. Specifically, *Sabottke*’s trained SVM classifiers are configured to compute class labels for exploitation, defining values for vulnerabilities as “either exploited or not exploited.” *Supra*, Claim 1(d); *see also Sabottke*, 6 (“[w]e compute...the class labels $Y \dots \{exploited, not\ exploited\}$ ” based on input features). *Sabottke*’s class labels are computed by applying the extracted features described above in Claims 1(c) and 1(e) and listed in figure 2 of *Sabottke* to *Sabottke*’s trained SVM classification model. *Sabottke*, Fig. 1; *supra*, Claims 1(c)

and 1(e) (describing CVSS Information, Twitter Traffic, and Database Information features).

Sabottke's class labels define one or more values (e.g., exploited or not exploited) that “defin[e] a likelihood of exploitation associated with the software vulnerability.” *Dec.*, ¶¶67-70. For example, SVMs output a prediction indicating which two classes a data point belongs to. *Dec.*, ¶69. For *Sabottke*'s exploit prediction, its SVM indicates whether a particular vulnerability is likely to be exploited, or not likely to be exploited. *Sabottke*, 13 (“Twitter data provides valuable insights into **the likelihood of exploitation**”), 15 (“**likelihood of real-world exploitation**, produced by our system”); *Dec.*, ¶¶69-70. Based on this, a vulnerability receives an exploitability class label that defines a binary value—“exploited, not exploited.” *Dec.*, ¶70. In this way, *Sabottke* computes class labels defining one or more values defining “which vulnerabilities are likely to be exploited [.]” *Id.*, 2 (“assessing the risk that the vulnerabilities will be exploited”); *Dec.*, ¶¶67-70.

8. ***Claim 1(f)(1): “wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.”***

Like the '897 Patent, *Sabottke* teaches predicting actual exploitation of software vulnerabilities before those vulnerabilities are disclosed to a known vulnerability database.

During prosecution, Applicant provided context for analyzing vulnerabilities “before disclosure.” In a July 13, 2023, request for continued examination after a final rejection, Applicant differentiated U.S. Patent Pub. No. 2017/0214710 to Seiver et al. (“Seiver”), stating:

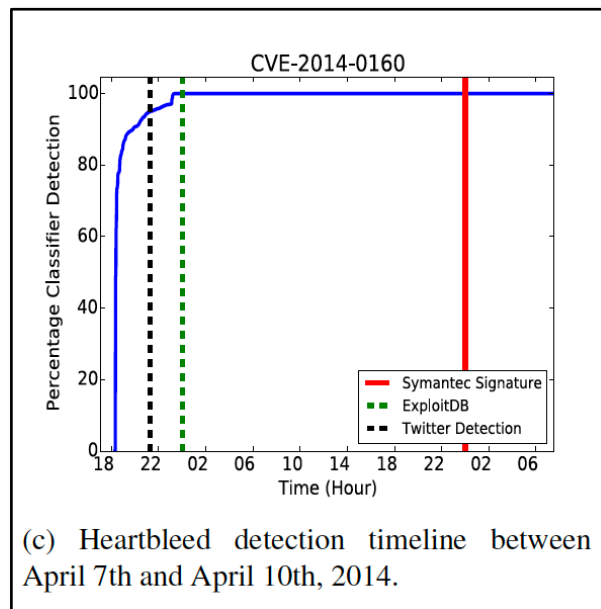
Seiver does not discuss predicting a likelihood of exploitation of a vulnerability discussed online prior to disclosure to a public database or prior to actual exploitation of the vulnerability; i.e., Seiver does not disclose the features of “wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data”

’897 File History, 480 (emphasis in original). Per Applicant, “Seiver is not directed to actively ‘scouting’ the web for vulnerabilities that have a high likelihood of exploitation and instead appears to be more directed towards defensive strategies for protecting a network.” *Id.*; see also *id.*, 480-81 (differentiating another reference based on this same limitation for not “consider[ing] text information associated with discussion of vulnerabilities by users posting hacking-related content at a web forum prior to disclosure to a public vulnerability database”). Like the claimed invention, *Sabottke* teaches actively scouting hacker communications in Twitter feeds to predict the likelihood a vulnerability will be exploited before disclosure to NVD or another vulnerability database. *Dec.*, ¶¶71-72.

Sabottke expressly teaches methods to “characterize the landscape of threats related to **information leaks about vulnerabilities before their public disclosure**”

to a public vulnerability database. *Sabottke*, 3. “For the vulnerabilities in our data set, Figure 3 compares the dates of the earliest tweets mentioning the corresponding CVE numbers with the public disclosure dates for these vulnerabilities. Using the disclosure date recorded in OSVDB, we identify 47 **vulnerabilities that were mentioned in the twitter stream before their public disclosure.**” *Id.*, 9 (“4.2 Information posted before disclosure”). Indeed, *Sabottke* teaches detecting the popular “Heartbleed” vulnerability before disclosure to the ExploitDB database. *Sabottke*, 13 (“Our Twitter-based classifier provides an ‘exploited’ output 3 hours after the first tweet appears related to this CVE on April 7, 2014” and before “Heartbleed appeared in ExploitDB on the day after disclosure (April 8, 2014)”).

Figure 8c illustrates this process:



Id., Fig. 8c. The dotted black line represents Twitter exploit detection and the dotted green line represents disclosure to the ExploitDB database. *Id.*, 13-14; *Dec.*, ¶73.

In the “non-goals” section of *Sabottke*, *Sabottke* explains its invention does not identify zero-day attacks which exploit vulnerabilities before those vulnerabilities are ever discussed in public forums. *Sabottke*, 3. In other words, *Sabottke* does not assess non-public vulnerabilities, like vulnerabilities mentioned in private feeds. *Id.* *Sabottke* does, however, look at vulnerabilities mentioned publicly (e.g., on twitter), but that have not been reported to public vulnerability databases like NVD or ExploitDB. *Sabottke*, Fig. 8c, 13-14. *Sabottke*’s teachings are consistent with the intrinsic record. *Dec.*, ¶¶73-74. The ’897 patent crawls hacker discussions on online forums but is interested in vulnerabilities mentioned on those forums that have not been disclosed to public vulnerability databases to predict whether that vulnerability will actually be exploited in the real world. *Dec.*, ¶75. *Sabottke* likewise predicts a likelihood of exploitation of vulnerabilities before disclosure to public vulnerability databases and therefore teaches this limitation. *Dec.*, ¶¶71-75.

B. Claim 3 - The method of claim 1, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.

Sabottke teaches accessing a plurality of datasets, including vulnerability data for publicly disclosed vulnerabilities (e.g., from NVD and OSVDB) for context for vulnerabilities mentioned in Twitter feeds. *Supra*, Claim 1(b). *Sabottke* also teaches

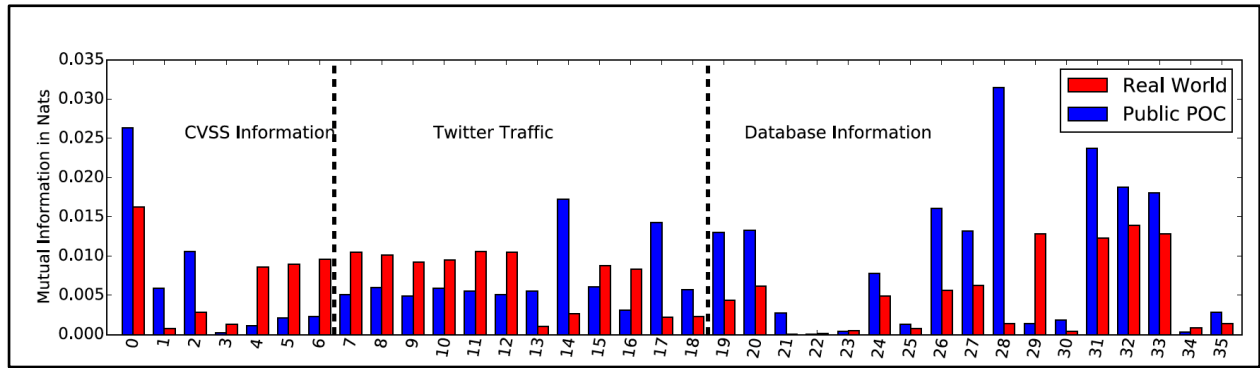
accessing exploits data for exploits that were used in real world attacks (e.g., from Symantec’s antivirus signatures and intrusion-protection signatures). *Sabottke*, 5. For real-world exploit data, *Sabottke* teaches its system “identif[ies] the set of vulnerabilities *exploited in the real world* by extracting the CVE IDs mentioned in the descriptions of Symantec’s anti-virus (AV) signatures [] and intrusion-protection (IPS) signatures.” *Id.* (emphasis in original); *see also id.*, 7 (Table 2 showing real-world exploits).

C. Claim 5 - The method of claim 1, wherein certain features correspond to a known vulnerability obtained from the plurality of datasets.

Sabottke teaches accessing certain features such as CVSS information and vulnerability database information from NVD and OSVDB corresponding to a known software vulnerability. *Supra*, Claim 1(e). These features correspond to “known vulnerabilities” such that they provide “context about the vulnerabilities discussed on Twitter” from vulnerability data previously disclosed in the public databases. *Sabottke*, 5; *Dec.*, ¶76.

D. Claim 9 - The method of claim 1, further comprising computing mutual information from the plurality of datasets informative as to what information a given feature provides about another feature.

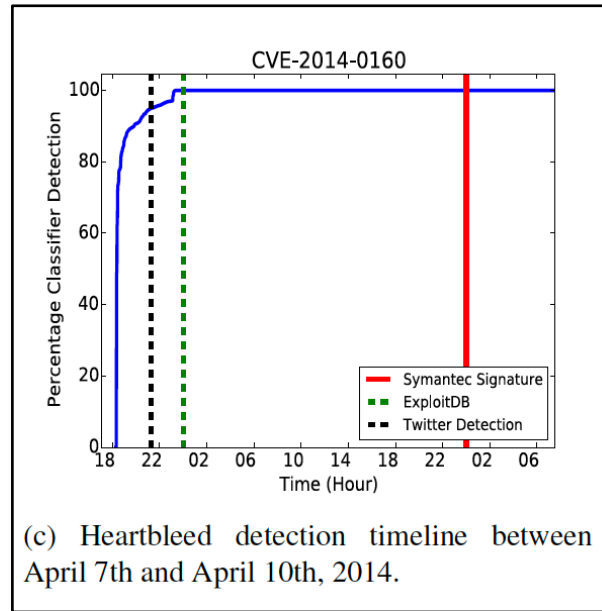
Sabottke teaches computing “[m]utual information” from the plurality of datasets:



Sabottke, Fig. 2; *see also id.*, 8 (“Figure 2: Mutual information between real world and public proof of concept exploits for CVSS, Twitter user statistics, NVD, and OSVDB features”). Mutual information (MI) is informative as to what information a given feature provides about another feature because it “can single out useful features suggesting that the vulnerability is exploited as well as features suggesting it is not.” *Sabottke*, 6; *Dec.*, ¶77.

E. Claim 10 - The method of claim 1, further comprising: detecting, from the plurality of datasets, vulnerabilities that appear before an associated exploitation date.

Sabottke teaches detecting vulnerabilities before an associated exploitation date by analyzing Twitter feeds for vulnerabilities mentioned that have yet to be exploited. *Sabottke*, 12-14. *Sabottke*’s classifier provides “early detection” for real-world exploits. *Id.* Figure 8c below “illustrates early detection for the case of Heartbleed (CVE-2014-1060)”:



Id., Fig. 8c. “The dashed black line represents the earliest time when our classifier is able to detect this vulnerability as a real-world exploit...3 hours after the first tweet appears related to this CVE” and well before a Symantec attack signature was generated (red line). *Id.*, 13 (describing detecting an exploit “within 10 minutes of its first appearance on Twitter”). *Sabottke* thus detects vulnerabilities mentioned on Twitter before an associated exploitation date. *Dec.*, ¶78.

F. Claim 14 - A computing device, configured via machine learning to apply a learned function to data associated with a software vulnerability to estimate a likelihood of exploitation of the software vulnerability, the learned function associated with a prediction model derived from at least one machine learning algorithm and a plurality of datasets associated with software vulnerabilities, the plurality of datasets including information associated with discussion of vulnerabilities by users posting hacking-related content, wherein the likelihood of exploitation predicts an actual exploitation of the software vulnerability before disclosure based on hacker communications from the plurality of datasets.

Sabottke teaches or suggests a computing device for effectuating the software functions described in claims 1 and 14. *Supra*, Claim 1 (describing software functions implemented by *Sabottke* and application by known hardware). The '897 Patent explains its claimed invention may be performed by an application hosted on or otherwise implemented by a “computing device 104” that “may include a server, controller, a personal computer, a terminal, a workstation...or other such computing device.” ’897 Patent, 6:11-19. Likewise, *Sabottke*’s described invention would have been implemented by a processor on a similar computing device. *Dec.*, ¶¶43-44, 79-80.

Sabottke applies supervised training algorithm to train its SVM machine learning prediction models with vulnerability data obtained from Twitter hacker discussions and vulnerability databases to estimate a likelihood of exploitation of the software vulnerability. *See generally*, Claim 1. The output from this prediction model predicts a likelihood of actual exploitation of the software vulnerability before

disclosure based on hacker communications from the Twitter feed. *Supra*, Claim 1(f)-1(f)(1).

G. Claim 15 - The computing device of claim 14, wherein the prediction model is at least one classification model that outputs from features of the software vulnerability a score indicative of the likelihood of exploitation.

Sabottke teaches its Twitter exploit prediction model utilizes an SVM classification model that outputs a likelihood of exploitation. *Supra*, Claim 1(f) (discussing computing likelihood of exploitation). *Sabottke* teaches or suggests outputting a score indicative of the likelihood of exploitation. For example, *Sabottke* teaches outputting “predictions” that are aggregated and compared to “SVM prediction thresholds.” *Sabottke*, 12-13. These predictions are “scores” in that they are values compared to a hyperplane. *Dec.*, ¶81 (SVM outputs serve “as a natural score for the exploitability of a vulnerability”).

H. Claim 18

1. Claim 18[Pre]: “A method of prioritizing vulnerabilities using cyber threat intelligence, comprising:”

To the extent the preamble is limiting, *Sabottke* teaches methods for prioritizing vulnerabilities using cyber threat intelligence. *Sabottke*, 2 (describing “detection techniques” for “response prioritization”), 3 (“Specifically, we investigate techniques for minimizing false-positive detections—vulnerabilities that are not actually exploited—which is critical for prioritizing response actions”), 7 (“we envision utilizing our classifier as a tool for prioritizing the response to

vulnerability disclosures”), 15 (“Twitter-based classifiers can be employed to guide the prioritization of response actions after vulnerability disclosures”).

2. ***Claim 18(a): “utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations including:”***

Sabottke teaches or suggests a processor in operable communication with memory that stores instructions for performing the claimed operations. *Supra*, Claim 1(a) (further noting a POSITA would have found it obvious to implement *Sabottke*’s “Twitter-based exploit detector” on a known computing system by executing software instructions in memory via a processor).

3. ***Claim 18(b): “accessing a plurality of datasets associated with a predetermined set of data sources, at least a portion of the plurality of datasets defining training data including hacker communications;”***

Sabottke teaches accessing a plurality of datasets from predetermined data sources (*Sabottke*, Fig. 1), including hacker communications from tweets. *Supra*, Claim 1(b).

4. ***Claim 18(c): “applying learning algorithms to the training data to generate a predictive model configured to predict a likelihood of exploitation of respective software vulnerabilities;”***

Sabottke teaches applying supervised learning algorithms to training data to generate an SVM predictive model configured to predict the likelihood of exploitation of software vulnerabilities. *Supra*, Claim 1(d).

5. ***Claim 18(d): “accessing one or more features associated with***

a software vulnerability; and”

Sabottke teaches accessing one or more features associated with a software vulnerability, including “CVSS Information” and “Database Information” features listed in Figure 2. *Supra*, Claim 1(e).

6. ***Claim 18(e): “computing, by applying the one or more features to the predictive model, one or more values defining a likelihood of exploitation associated with the software vulnerability, wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.”***

Sabottke teaches or suggests computing a value defining a likelihood of exploitation associated with the software vulnerability using its trained SVM predictive model. *Supra*, Claim 1(f). This value predicts an actual exploitation before disclosure to a public database based on hacker communications from the training data. *Supra*, Claim 1(f)(1).

I. Claim 20 - The method of claim 18, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.

Sabottke teaches accessing a plurality of datasets, including vulnerability data for publicly disclosed vulnerabilities (e.g., from NVD and OSVDB) and exploits data for exploits that were used in real world attacks (e.g., from Symantec’s antivirus signatures and intrusion-protection signatures). *Supra*, Claim 3.

VI. GROUND 2: CLAIMS 1-3, 5, 10-11, 13-15, 17-20 ARE OBVIOUS OVER SMYTH AND PHILLIPS

A. Claim 1

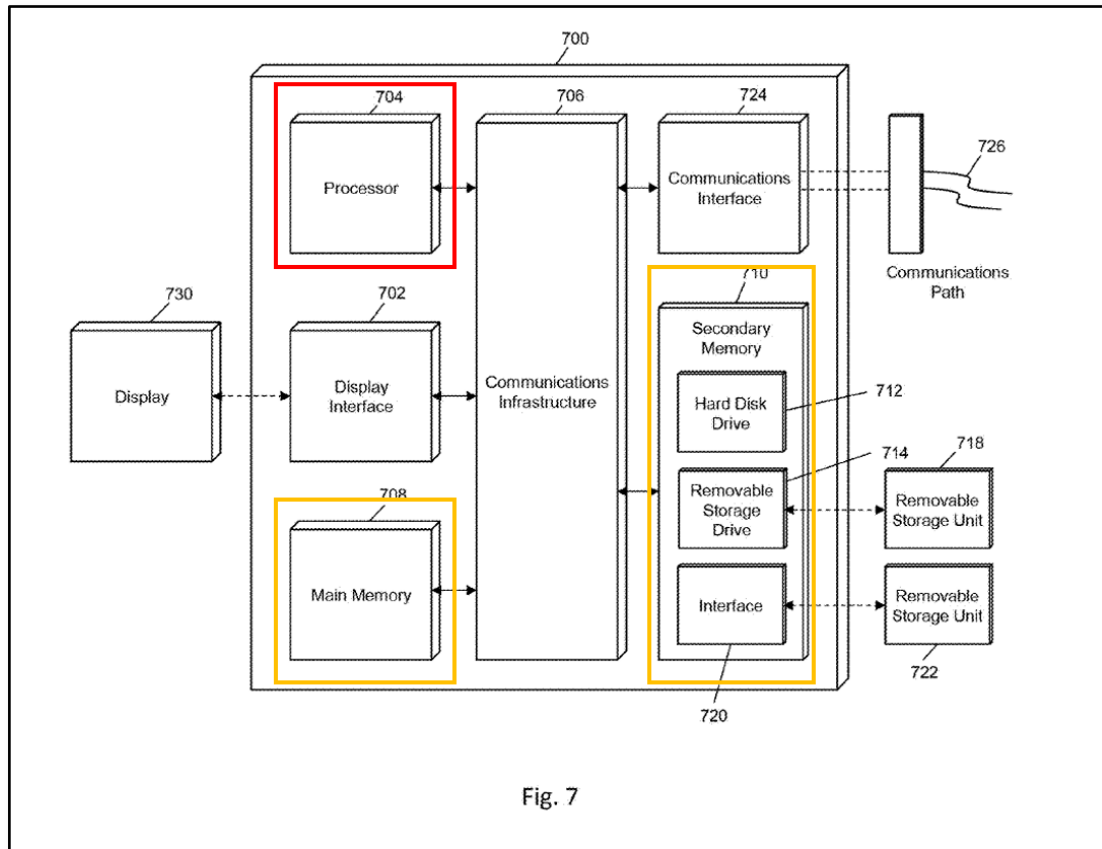
1. Claim 1(Pre): “A method for assessing a likelihood of exploitation of software vulnerabilities, comprising:”

To the extent the preamble is limiting, *Smyth* teaches methods for using a “predictive engine for analyzing existing vulnerability information to determine the **likelihood of a vulnerability being exploited** by malicious actors against a particular computer or network of computers.” *Smyth*, 1:7-11, 2:25-50, 3:63-67, 4:4-34, 11:22-25, claims 1 and 10. *Smyth* teaches “malicious actors may use **vulnerabilities in existing software**, hardware, or combinations of software and hardware” to compromise the integrity of computers and networked computer systems. *Id.*, 1:19-24. *Smyth*’s methods pair data input to the predictive engine with particular vulnerabilities to determine a likelihood of exploitation. *Id.*, 5:40-6:10. This includes assessing the likelihood of exploitation of “[v]**ulnerable software and versions**[.]” *See, e.g., id.*, Figure 3.

2. Claim 1(a): “utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations, including:”

Smyth teaches a “processor unit” that performs the steps of its described methods. *Smyth*, 4:4-34, 6:17-9:17, 12:30-60, Figs. 1, 5, 6, 7. Processor 704, boxed in red, is depicted below in operable communication with main memory 708 and

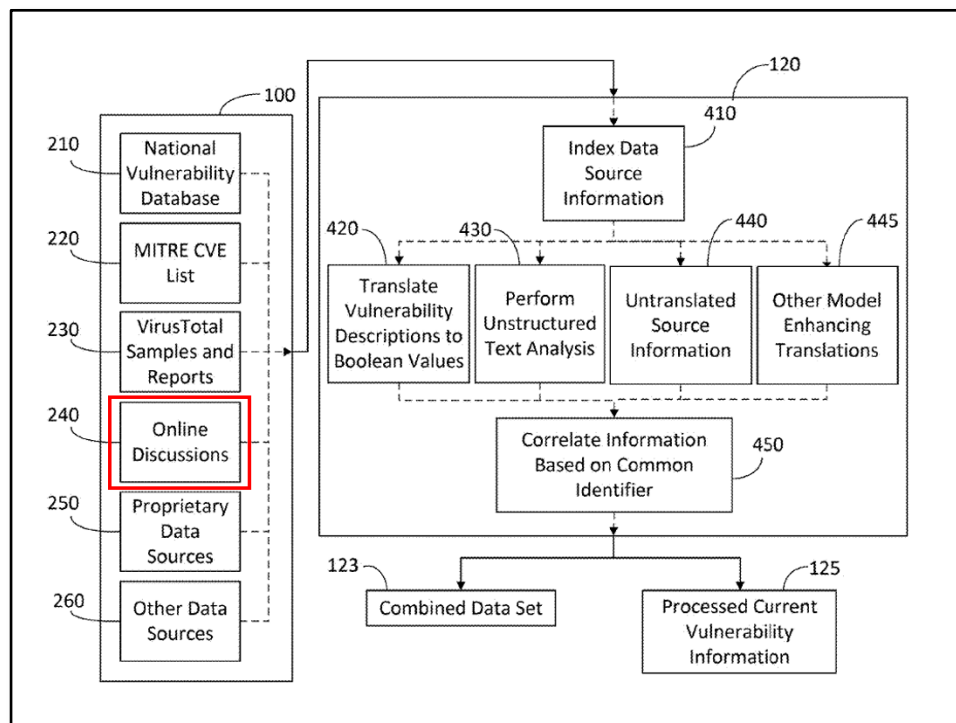
secondary memory 710, boxed in yellow, through communications infrastructure 706:



Id., Fig. 7 (annotated). The processor contains “modules or engines” that “utilize software, such as [] program code and/or programs stored in the main memory 708 or secondary memory 710” “to perform the functions discussed” in *Smyth. Id.*, 12:30-60; *see also id.*, 11:37-42 (“the method 100 of FIG. 1 may be implemented in the computer system 700 using...instructions for performing the tasks described for the method 100”).

3. ***Claim 1(b): “accessing a plurality of datasets associated with a predetermined set of data sources, the plurality of datasets including training data comprising hacker communications;”***

Smyth's processor accesses a plurality of datasets from multiple predetermined data sources. *Smyth*, Abstract, 4:4-24. Namely, “data processor 120 which executes on processor device 704” processes information obtained from data sources 110 “so that a combined data set 123 of historical vulnerability information is obtained” along with “current vulnerability information” 125. *Id.*, 14:33-43, 14:61-65. “The information from the data sources 110 may be accessed by the data processor 120 through application programming interface (API) calls, web page scraping, periodic database imports, or other known techniques.” *Id.*, 5:49-52. This process is shown in Figure 4 below:



Id., Fig. 4 (annotated); *see also id.*, Figs. 1-2.

Smyth accesses multiple datasets from the predetermined data sources identified in Figure 4 to construct a “combined data set” used to train predictive models. *Id.*, 4:6-14 (describing training prediction models 531-534 using the combined data set). This data includes “online discussions 240,” boxed in red. *Supra*, Fig. 4. “[O]nline discussions 240” include “those on social media and discussion forums.” *Id.*, 4:44-47, 6:30-36. While *Smyth* discloses accessing training data from online “discussion forums[,]” *Smyth* does not expressly discuss accessing “hacker communications” in those forums. *Phillips* teaches this.

Phillips teaches “crawling a variety of radical websites” to collect forum data from forums associated with the “Dark Web Forum Portal[.]” *Phillips*, 3. Table II below provides an overview of exemplary dark web forums:

Forum	#Messages:	# Threads:	# Members:	Start Date:	End Date:	Forum URL:
Ansar AlJihad Network (AnsarI)	29492	11244	382	12/08/08	01/20/2010	http://www.ansarI.info/
Gawaher (Gawaher)	372499	53235	9269	10/24/2004	06/07/12	http://www.gawaher.com
Islamic Awakening (IslamicAwakening)	201287	32879	3964	04/28/2004	05/22/2012	http://forums.islamicawakening.com
Islamic Network (IslamicNetwork)	91874	13995	2082	06/09/04	11/10/10	http://talk.islamicnetwork.com
Islamic Web-Community (Myiwc)	25016	6310	756	11/05/00	02/19/2010	http://www.myiwc.com/forums/index.php
Turn To Islam (TurnToIslam)	335338	41654	10858	06/02/06	05/20/2012	http://www.turnintonislam.com/forum/
Ummah	1491957	91527	21013	04/01/02	05/18/2012	http://www.ummah.com/forum/

Id., 4. A POSITA would have understood dark web forums include “hacker communications[.]” *Dec.*, ¶¶48-49, 95-99 (noting the hacker communications in the ’897 Patent were crawled from the dark web); Ex. 1015 (Macdonald), 2 (“hacker communities are dark networks”); Ex. 1016 (CTC Sentinel), 1 (describing “Jihadist Hackers” groups communicating in online forums); Ex. 1013 (Benjamin), 4

(“Hackers often congregate within underground communities, most commonly in the form of...online forums”); Ex. 1017 (Guetler), 90 (describing a few works that “have focused on Dark Web forums of terrorist and hacker groups” including *Phillips*). It would have thus been obvious to crawl dark web forums, as disclosed in *Phillips*, for hacker communications for use in *Smyth*’s predictive model. *Dec.*, ¶99 (explaining the examiner during prosecution found crawling dark web forums for hacker communications was admittedly known in AAPA).

A POSITA would have been motivated to crawl dark web forums in *Smyth* pursuant to *Phillips* to provide a rich repository of current hacker communications for *Smyth*’s predictive engine to analyze. *Dec.*, ¶100; Ex. 1013 (Benjamin), 4 (“Malicious software, hacking tutorials, and other resources intended to help conduct cybercrime can be commonly found within hacker communities” on forums). This would facilitate *Smyth*’s goal to obtain existing threat information such that *Smyth*’s system can predict the likelihood of exploitation of “newly discovered” vulnerabilities. *Smyth*, 2:6-21, 14:53-57; *see also id.*, 6:33-34 (“more current information [] may be found in online discussions 240”); *Dec.*, ¶100.

A POSITA would have had a reasonable expectation of success modifying *Smyth* to crawl dark web forums for hacker communications as taught by *Phillips*. *Dec.*, ¶¶101-102. *Smyth* already teaches analyzing “online discussions 240”/“discussion forums” to determine the likelihood of a vulnerability being

exploited. *Smyth*, 4:35-47, Fig. 2. *Smyth* would thus utilize the same hardware to crawl dark web forums containing hacker communications to effectuate its methods to predict the likelihood of exploitation of vulnerabilities. *Dec.*, ¶101. *Smyth* also explains “[o]ther data sources 260 may also be considered” and “[e]ach of the data sources 210, 220, 230, 240, 250, 260 shown in Fig. 2 are merely examples and do not limit the number of type of data sources which may be considered by the method 100 described.” *Smyth*, 4:47-52. *Smyth* therefore contemplates examining other sources of data and it would have been well within the skill of a POSITA to modify *Smyth* to examine hacker communications in dark web forums. *Dec.*, ¶102. Crawling and analyzing dark web forums was a well-known strategy for early detection of new emerging cyber threats, and a POSITA would have had success implementing that known strategy in *Smyth*’s system already equipped to analyze discussion forums. *Dec.*, ¶¶48-49, 101-102; Ex. 1015 (Macdonald), 1 (describing analyzing the “communication activity” of hackers on forums to identify threats); *supra*, Sec. II.B (’897 File History) (discussing AAPA for crawling hacker communications on forums).

4. Claim 1(c): “accessing features from the plurality of datasets that include measures computed from social connections of users posting hacking-related content;”

Smyth teaches accessing features from its plurality of datasets using several feature extraction techniques. For example, *Smyth* teaches using “unstructured text

analysis 430” “on certain portions of the vulnerability information so that features may be derived.” *Smyth*, 7:22-24. “[O]nline discussions 240 may include forum posts formed substantially of unstructured text.” *Id.*, 7:28-30. This “unstructured text is analyzed [] to facilitate consideration by the predictive models[.]” *Id.*, 7:33-35; *see also id.*, 6:18-42 (“information may be derived from the data sources 110 for use in the combined data set 123 and the processed current vulnerability information 125”).

While *Smyth* teaches feature extraction techniques, *Smyth* does not expressly discuss accessing features that include “measures computed from social connections of users posting hacking-related content[.]” *Phillips*, however, teaches using “Social Network Analysis (SNA) techniques” to access social measures in dark web forums that overlap with measures described in the ’897 Patent’s specification.

Namely, *Phillips* uses the SNA metrics in Table I below to access measures computed from user’s posting in dark web forums:

TABLE I. A TABLE OUTLINING THE METRICS USED TO EVALUATE OUR SOCIAL NETWORK	
Attribute Name	Description
Sent Messages (SM)	The number of emails sent by an employee.
Received Messages (RM)	The number of emails received by an employee.
Degree Centrality (DCS)	The number of distinct employees within the network that an employee has sent emails to.
Betweenness Centrality Score (BCS)	The betweenness centrality measure for an employee.[20]
Pagerank Score (PRS)	The PageRank score an employee. [21]
Markov Ranking (MR)	The markov ranking of an employee. [22]
HITS Authority Score (HAS)	The authority score for an employee (if several users with high hub weights send an email t the user then they will have a higher authority score). [23]
HITS Hub Score (HHS)	The hub score for an employee (if the user sends emails to users with high authority scores then they will have a higher hub score). [23]
Clique Score (CS)	The number of cliques (maximal subgraphs) an employee is in using the Bron and Kerbosch algorithm.[24]
Weighted Clique Score (WCS)	The weighted clique score for each user, weighted by the number of users within each clique.
Average Distance Score (ADS)	The average distance between the user and all other users in the graph.
Clustering Coefficient (CC)	The extent to which vertices in a graph tend to cluster together. [25]

Phillips, 3 (annotated). The highlighted features in Table I above substantively overlap with the social network measures discussed in the '897 Patent. *Compare* '897 Patent, 21:6-46 (“[i]n degree[,]” “[o]ut-degree[,]” “[i]n-degree centrality[,]” “[o]ut-degree centrality[,]” “[b]etweenness centrality[,]” “[p]agerank centrality”).

The SNA metrics in Table I are the “top-ranked SNA metrics from previous work” by the authors of *Phillips*. *Phillips*, 2. *Phillips* utilizes those top-ranked metrics on “the forums collected as part of the Dark Web Forum Portal dataset.” *Id.*, 3; *see also id.*, Abstract (“For this experiment, we conducted our investigation on five forums collected in previous work as part of the Dark Web Forum portal and built upon the tool support created in our previous research in order to visualise and analyse the network” by utilizing “SNA metrics” on the “communication metadata” from these dark web forums). *Phillips* “first calculate[es] a range of SNA metrics to better understand the group members” in dark web forums and “then appl[ies] unsupervised learning in order to create clusters that would help classify the Dark Web Forums users[.]” *Id.*; *see also id.*, 1 (“we set out to trial various Social Network Analysis (SNA) techniques in order to identify a range of potentially ‘important’ members of the [dark web] forums”).

In this process, *Phillips* teaches collecting dark web forum posts, extracting metadata from these posts, building a social network, and then analyzing that social network with SNA techniques and machine learning. *Phillips*, 1-2 (*see* “II.

Methodology”); *Dec.*, ¶¶103-108(describing the “experimental setup”). And as described above, a POSITA would have understood certain dark web users post hacking-related content. *Supra*, Claim 1(b). *Phillips* therefore analyzes hacking-related content in dark web posts. *Dec.*, ¶¶99, 108.

A POSITA would have been motivated to modify *Smyth* to implement *Phillips*’ techniques for identifying important members on forums to improve *Smyth*’s methods for exploit prediction. *Dec.*, ¶109. Correlating exploit and vulnerability information from discussion forums with social measures from the user posting in the forum would improve *Smyth*’s threat predictions. *Dec.*, ¶109. *Smyth*’s system generates threat levels for current vulnerabilities. *Smyth*, 4:29-34, 7:7-8, 11:26-33, 15:8-14. Assessing social measures related to the user posting in the forum would improve threat level generation such that a vulnerability discussed in a post from an “important” user would generate a higher threat level than a post from a lesser important user. *Dec.*, ¶109. This would have provided additional features for analysis in *Smyth*’s system and would have increased the accuracy of *Smyth*’s threat predictions. *Dec.*, ¶109.

A POSITA would have had a reasonable expectation of success modifying *Smyth* to utilize *Phillips*’ SNA metrics to access features computed from social connections of users posting hacking-related content. *Dec.*, ¶¶110-111. *Smyth* already teaches feature extraction techniques that extract features from discussion

forum data to feed its predictive models. *Smyth*, 7:22-35. And *Smyth* expressly states its embodiments “may employ other techniques not specifically described here to take the information from the data sources 110 and provide it in a manner suitable for the prediction engine 160.” *Id.*, 6:39-42; *see also id.*, 8:7-10. Thus, it would have been well within the skill of a POSITA to incorporate other well-known techniques, like *Phillips*’ SNA metrics, in *Smyth*’s system to process the dark web forum data for *Smyth*’s predictive models. *Dec.*, ¶¶109-111.

5. ***Claim 1(d): “applying learning algorithms to the training data to generate classification models that are configured to predict class labels defining a likelihood of exploitation of respective software vulnerabilities;”***

Smyth teaches applying various learning algorithms (e.g., 531-534) to the training data to generate prediction ensemble 155 configured to predict class labels defining a likelihood of exploitation of software vulnerabilities. *Smyth*’s training and model generation process is illustrated in Figure 5 below:

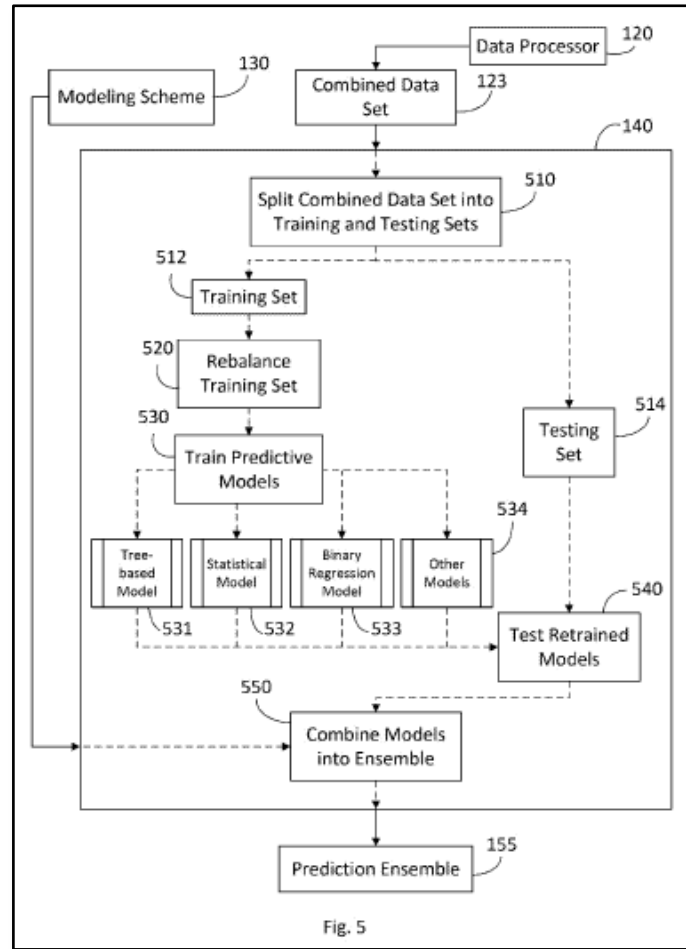


Fig. 5

Smyth, Fig. 5. As shown, *Smyth* trains learning algorithms 531-534 with data from its combined data set 123. *Id.*, 9:20-26, 10:25-27. Combined data set 123 includes data from at least a portion of the data sources, including data captured from online discussions 240. *Id.*, 10:5-12 (“At least a portion of the combined data set 123 is relied upon for retraining”), Fig. 2 (capturing data from online discussions 240), Fig. 4 (combined data set 123 generated in part from online discussions data 240). In *Smyth*’s modified system, this includes training data captured from hacker communications as taught by *Phillips*. *Supra*, Claim 1(b). Thus, *Smyth-Phillips*

teaches applying learning algorithms to training data comprising hacker communications. *Dec.*, ¶112.

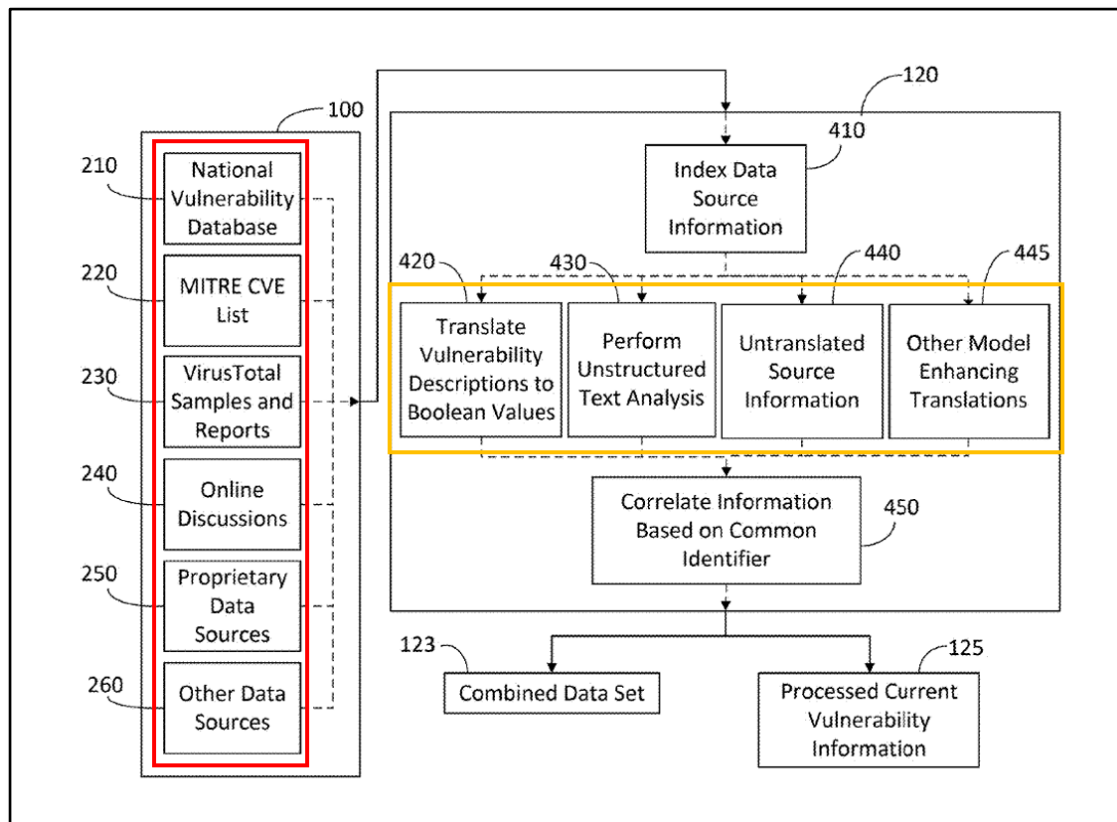
After training, learning algorithms 531-534 are combined into prediction ensemble 155. *Supra, Smith*, Fig. 5. Prediction ensemble 155 comprises trained algorithms 531-534 and therefore comprises “classification models” as claimed. *Dec.*, ¶113. The ’897 Patent provides exemplary classification models that predict “class labels.” ’897 *Patent*, 14:30-33, 17:64-65, 21:57-59. Some example classification models are Decision Trees, Naïve Bayes classifiers, Logistic Regressions, Support Vector Machines, and Bayesian Networks. *Id.*, 22:66-23:32. *Smyth* discloses overlapping classification models including “a tree-based model 531, a statistical model 532, a binary regression model 533, and other models relying on other algorithms 534[.]” *Smyth*, 9:42-45.

Smyth’s models are combined into ensemble 155 which “generate a prediction 170” that “provides a re[pre]sentation of the likelihood of the current vulnerability will [be] exploited by malicious actors against a particular computer network.” *Smyth*, 11:18-25. *Smyth* therefore teaches “classification models” (e.g., prediction ensemble 155) that predict “class labels” (e.g., exploitability) “defining a likelihood of exploitation of respective software vulnerabilities” (e.g., prediction 170). *Dec.*, ¶¶114-115, 118.

6. **Claim 1(e): “accessing one or more features associated with a software vulnerability; and;”**

Smyth teaches accessing features associated with a software vulnerability, including features from NVD, the MITRE CVE List, VirusTotal Samples and Reports, Online Discussions, Proprietary Data Sources, and Other Data Sources.

Smyth, 4:36-55. This process is shown below in Figure 4 below:

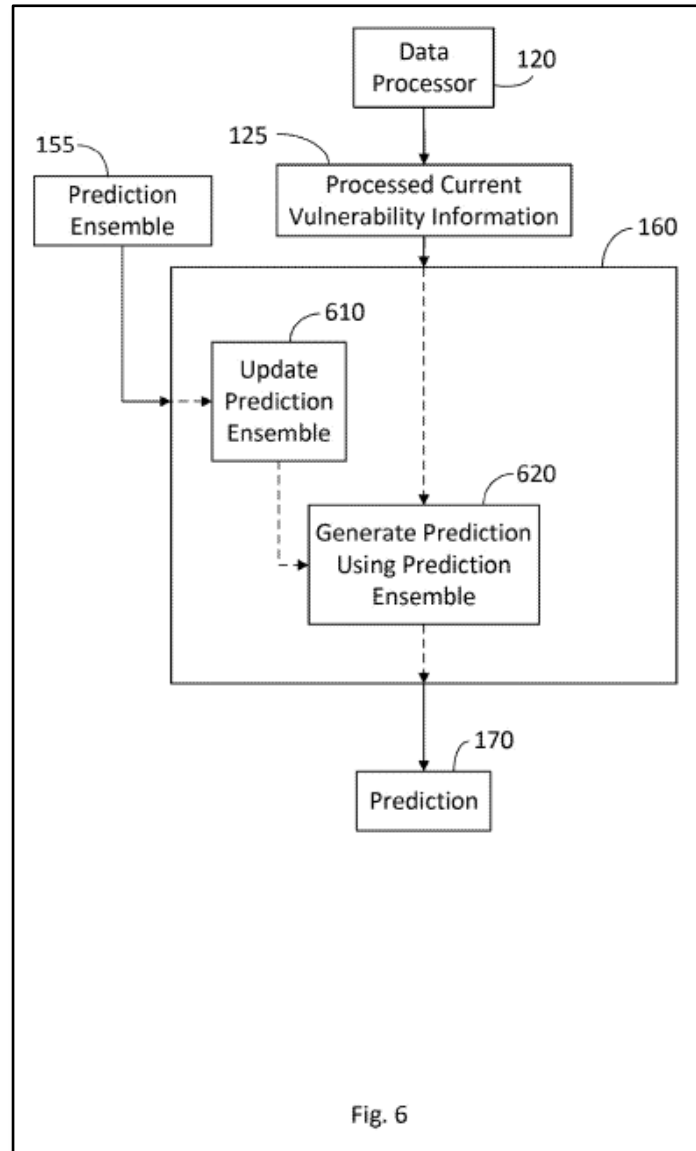


Smyth, Fig. 4 (annotated). Feature extraction techniques 420-445 (orange) extract features from datasets 210-260 (red) that are associated with a software vulnerability. Namely, *Smyth* teaches “data processor 120 creates processed current vulnerability information 125 by obtaining information about a current vulnerability

from the data sources[.]” *Id.*, 4:21-24, 6:18-42 (“certain information may be derived from the data sources 110” including “current vulnerability information 125”). This includes NVD features such as “vulnerability descriptions” or “CVSS” base score. *Id.*, 6:56-62, 7:18-21, Fig. 3. These features overlap with vulnerability features described in the ’897 Patent. *See* ’897 Patent, Fig. 1B (“NVD” features), 6:60-67 (“features include...vulnerability description”), 6:67-7:1 (“Additional features are included from NVD namely, CVSS score[.]”).

7. *Claim 1(f): “computing, by applying the one or more features to the classification model, a class label defining one or more values defining a likelihood of exploitation associated with the software vulnerability”*

Smyth teaches or suggests a prediction engine 160 that applies one or more of the above current vulnerability features to prediction ensemble 155 to compute a class label (e.g., exploitability) defining one or more values (e.g., exploited; not exploited); defining a likelihood of exploitation (e.g., prediction 170). *Supra*, Claim 1(d); *Dec.*, ¶¶112-115, 117. This process is illustrated in Figure 6 below:



Id., Fig. 6. Processed current vulnerability information 125 includes the features derived from the datasets described above. *Supra*, Claim 1(e). This includes “information from the data sources 100 that include more current information such as online discussion 240[.]” *Smyth*, 11:3-7. *Smyth* teaches “[t]he prediction engine 160 takes [] processed current vulnerability information 125 and uses the prediction

ensemble 155...to generate a prediction 170 of the likelihood the current vulnerability may be employed[.]” *Id.*, 4:24-29, 10:61-64, 11:18-25.

Smyth’s prediction ensemble computes the exploitability of a vulnerability. *Dec.*, ¶118. The class label is therefore the variable “exploitability” and that variable is defined by values such as “exploited” or “not exploited.” *Dec.*, ¶118. This is consistent with the ’897 Patent’s disclosures. ’897 Patent, 17:63-65 (“variable (here the class label y^c {exploited, not exploited})”). The exploitability of a vulnerability, e.g., its likelihood to be exploited, is defined by prediction 170. *Smyth*, 11:22-25 (“prediction 170 provides a representation of the likelihood [] the current vulnerability will [be] exploited”); *Dec.*, ¶118.

8. *Claim 1(f)(1): “wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.”*

Smyth-Phillips teaches this limitation. As explained in Ground 1, during prosecution, Applicant explained predicting vulnerabilities “before disclosure” refers to actively scouting the web for vulnerabilities not yet disclosed to public vulnerability databases. *Supra*, Ground 1, Claim 1(f)(1).

Unlike the references cited during prosecution, *Smyth* expressly scouts text information from “online discussions 240” as part of current vulnerability information 125 to generate a prediction on how likely that vulnerability will be exploited. *Supra*, Claim 1(b). As stated, “prediction engine 160 generates a

prediction based on processed current vulnerability information 125[.]” *Smyth*, 10:61-64. “[C]urrent vulnerability information 125 contains vulnerability information about a threat that is newly discovered or currently spreading through computer networks” and includes “more current information such as online discussions 240[.]” *Id.*, 11:1-8; *see also id.*, 14:55-57 (“The current vulnerability may be newly discovered, newly relied upon by malicious actors, or the like”). In other words, *Smyth*’s prediction engine 160 proactively scouts online discussions for newly discovered vulnerabilities that have not yet been added to a public database or not yet exploited. *Dec.*, ¶¶119-122.

In *Smyth*’s modified system, *Smyth* scouts online discussions from dark web forums that include hacker communications, and uses those hacker communications as training and testing data for *Smyth*’s predictive model. *Supra*, Claim 1(b) (proposed modification with *Phillips*). Thus, *Smyth-Phillips* teaches prediction 170 predicts an actual exploitation of the software vulnerabilities before disclosure based on hacker communications from the training data. *Dec.*, ¶¶119-123.

B. Claim 2 - The method of claim 1, further comprising generating a plurality of estimation outputs based on the one or more values to derive an overall quantitative score.

Smyth teaches its prediction ensemble generates a plurality of estimation outputs from multiple algorithms and uses those outputs to derive an overall prediction 170. Namely, *Smyth* teaches “prediction ensemble 155” uses different

modeling schemes “to tailor the predictions 170 in a manner that provides a more useful prediction 170.” *Smyth*, 9:46-54. An example modeling scheme includes “stacking” the different algorithms in the ensemble to generate “predictions of multiple algorithms[.]” *Id.*, 10:57-58, 11:57-59. These plurality of predictions from the multiple algorithms in prediction ensemble 155 “generate a prediction 170 of the likelihood the current vulnerability may be employed[.]” *Id.*, 4:24-28, 11:18-21 (“the predictive models 531, 532, 533, 534” are executed “in accordance with the prediction ensemble 155 to generate a prediction 170”). Prediction 170 can be “converted into a threat level” based on defined thresholds, such as a low, medium, or high threat. *Id.*, 4:29-34, 7:6-8, 11:26-33.

Prediction 170 is derived from the multiple estimation outputs of the various algorithms in ensemble 155 and represents the overall likelihood the current vulnerability will be exploited. *Id.*, 11:18-25. Prediction 170 is compared to thresholds to define a threat level and is therefore an “overall quantitative score” as claimed. *Dec.*, ¶¶124-126.

C. Claim 3 - The method of claim 1, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.

Smyth teaches accessing a plurality of datasets, including vulnerability data for publicly disclosed vulnerabilities (e.g., from the public National Vulnerability Database, MITRE CVE List, Other Data Sources, etc.) and exploits data that

includes exploits used in real world attacks (e.g., from VirusTotal Samples and Reports and Online Discussions). *Smyth*, 4:35-55, Fig. 2.

Regarding vulnerability data for vulnerabilities that are publicly disclosed, *Smyth* teaches “NVD 210” is “a publicly available data source” that is based on the CVE list 220, and list 220 “serves [as] a list or dictionary of CVE identifiers for publicly disclosed vulnerabilities.” *Id.*, 4:40-42, 4:56-58. NVD and the MITRE CVE list provide a public repository for publicly disclosed vulnerabilities. *Dec.*, ¶¶127-128. Indeed, the ’897 Patent confirms that NVD “maintains a database of publicly disclosed vulnerabilities each one identified using a unique CVE-ID.” ’897 Patent, 7:47-49.

Regarding exploits data for exploits used in real world attacks, *Smyth* teaches accessing data from VirusTotal Samples and Reports and online discussions (*Smyth*, 4:44-47), both of which include real world exploit data. *Dec.*, ¶129. *Smyth* teaches VirusTotal “provides a service for analyzing suspicious files and URLs, and for detecting viruses, worms, Trojans, and other kinds of malware.” *Smyth*, 5:1-9. VirusTotal is a website that allows users to submit files or URLs that are analyzed by antivirus scanners to identify real-world security threats, including exploits. *Dec.*, ¶129; Ex. 1018 (VirusTotal.com screen capture) (VirusTotal lets you “see which CVE are being exploited by attackers in the wild at this moment or during the last few weeks, what kind of malware they are distributing, and what actors are behind”);

id., (providing data on malware with “tag:exploit”). VirusTotal therefore includes real-world exploit data. *Dec.*, ¶129. As do online discussions. *Smyth*, 7:28-33 (online discussions can show if “a vulnerability is being actively being used”), 11:1-3 (online discussions contain information “about a threat that is newly discovered or currently spreading through computer networks”); *Dec.*, ¶¶129-130.

D. Claim 5 - The method of claim 1, wherein certain features correspond to a known vulnerability obtained from the plurality of datasets.

As discussed in Claim 1(e), *Smyth* teaches accessing certain features corresponding to a known software vulnerability, including features from NVD, the MITRE CVE List, VirusTotal Samples and Reports, Online Discussions, Proprietary Data Sources, and Other Data Sources. *Supra*, Claim 1(e). Feature extraction techniques 420-445 extract features from datasets 210-260 that are associated with a software vulnerability. *Smyth*, 4:21-24, 6:18-42 (“certain information may be derived from the data sources 110” including “vulnerability information 125”). Certain features correspond to a “known vulnerability[,]” such as, for example, NVD “vulnerability descriptions” or “CVSS” base score. *Id.*, 6:56-62, 7:18-21, Fig. 3. These are features from previously disclosed or “known” vulnerabilities. *Dec.*, ¶131; ’897 *Patent*, 4:31-40 (NVD comprises a comprehensive list of vulnerabilities disclosed, including a CVSS score), 10:1-14 (Table 2 – NVD description).

E. Claim 10 - The method of claim 1, further comprising: detecting, from the plurality of datasets, vulnerabilities that appear before an associated exploitation date.

Smyth detects vulnerabilities that appear before an associated “exploitation date” by analyzing online discussion data for vulnerabilities mentioned, but not yet exploited. The ’897 Patent defines an “exploitation date of a vulnerability” “as the date it was first detected in the wild.” ’897 *Patent*, 9:42-43, 11:46-48 (vulnerability mentioned online before exploitation date), 12:29-30 (same).

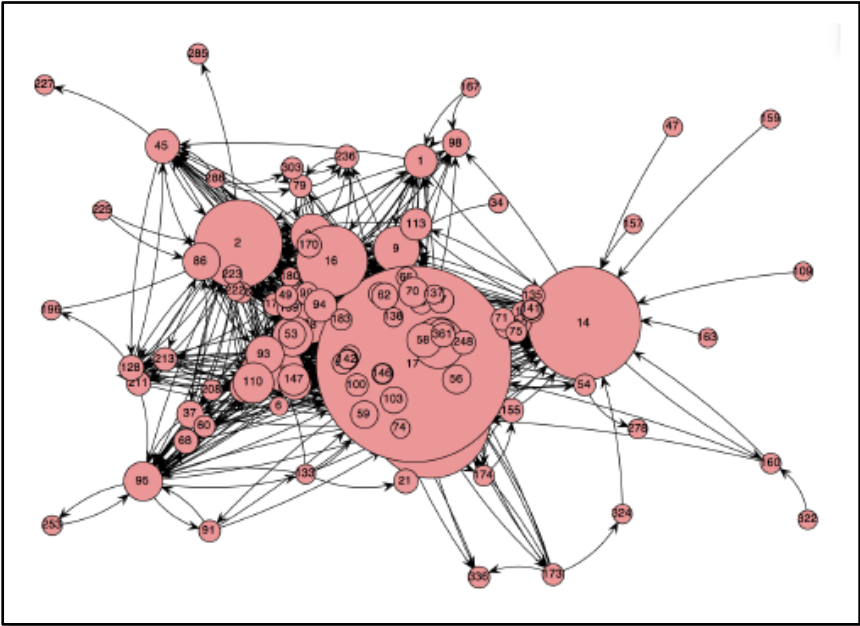
As discussed in Claim 1(f), *Smyth* teaches that “current vulnerability information 125 contains vulnerability information about a threat that is newly discovered” and includes “more current information such as online discussions 240[.]” *Supra*, Claim 1(f); *Smyth*, 11:1-8; *see also id.*, 14:55-57 (“The current vulnerability may be newly discovered, newly relied upon by malicious actors, or the like”). Thus, *Smyth* scouts online discussions for vulnerabilities mentioned that “newly discovered” and have not yet been exploited. *Dec.*, ¶¶132-133.

F. Claim 11 - The method of claim 1, further comprising: accessing features from the plurality of datasets that measure a centrality of the users in a social graph.

Smyth-Phillips teaches accessing features from the plurality of datasets that measure a centrality of users in a social graph. *Phillips* constructs a social graph of users posting in forums and measures graph properties using SNA metrics, including “centrality” features. *Phillips*, 2-3 (“in order to measure the graph properties of each

node that reflect importance, we used the SNA metrics outlined in...Table I”).

Phillips provides an example social graph, reproduced below:



Id., 3. Phillips uses the below centrality features to measure the graph properties of this social network:

TABLE I. A TABLE OUTLINING THE METRICS USED TO EVALUATE OUR SOCIAL NETWORK	
Attribute Name	Description
Sent Messages (SM)	The number of emails sent by an employee.
Received Messages (RM)	The number of emails received by an employee.
Degree Centrality (DCS)	The number of distinct employees within the network that an employee has sent emails to.
Betweenness Centrality Score (BCS)	The betweenness centrality measure for an employee.[20]
Pagerank Score (PRS)	The PageRank score an employee. [21]
Markov Ranking (MR)	The markov ranking of an employee. [22]
HITS Authority Score (HAS)	The authority score for an employee (if several users with high hub weights send an email t the user then they will have a higher authority score). [23]
HITS Hub Score (HHS)	The hub score for an employee (if the user sends emails to users with high authority scores then they will have a higher hub score). [23]
Clique Score (CS)	The number of cliques (maximal subgraphs) an employee is in using the Bron and Kerbosch algorithm.[24]
Weighted Clique Score (WCS)	The weighted clique score for each user, weighted by the number of users within each clique.
Average Distance Score (ADS)	The average distance between the user and all other users in the graph.
Clustering Coefficient (CC)	The extent to which vertices in a graph tend to cluster together. [25]

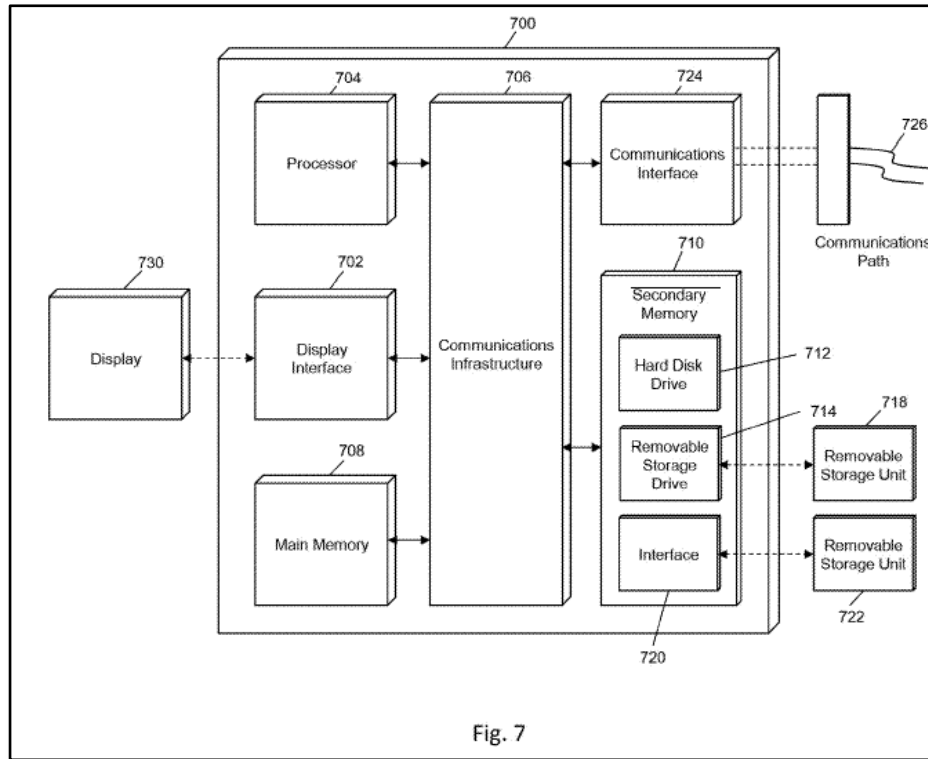
Id. (annotated).

G. Claim 13 - The method of claim 1, further comprising vectorizing text features derived from textual content of the plurality of datasets using a predetermined natural language process (NPL).

Smyth teaches processing unstructured text in online forum discussions using a predetermined natural language process such as latent Dirichlet allocation (LDA). *Smyth*, 7:22-47 (“the unstructured text may be processed by generative statistical models such as latent Dirichlet allocation (LDA)”). LDA is a natural language process that analyzes vectorized text features to “determine the relationships between topics contained in the unstructured text in a manner that may be considered by the prediction models[.]” *Smyth*, 7:40-43; *Dec.*, ¶134.

H. Claim 14 - A computing device, configured via machine learning to apply a learned function to data associated with a software vulnerability to estimate a likelihood of exploitation of the software vulnerability, the learned function associated with a prediction model derived from at least one machine learning algorithm and a plurality of datasets associated with software vulnerabilities, the plurality of datasets including information associated with discussion of vulnerabilities by users posting hacking-related content, wherein the likelihood of exploitation predicts an actual exploitation of the software vulnerability before disclosure based on hacker communications from the plurality of datasets.

Smyth teaches a computing device, depicted below, configured to perform the functions described in claims 1 and 14:

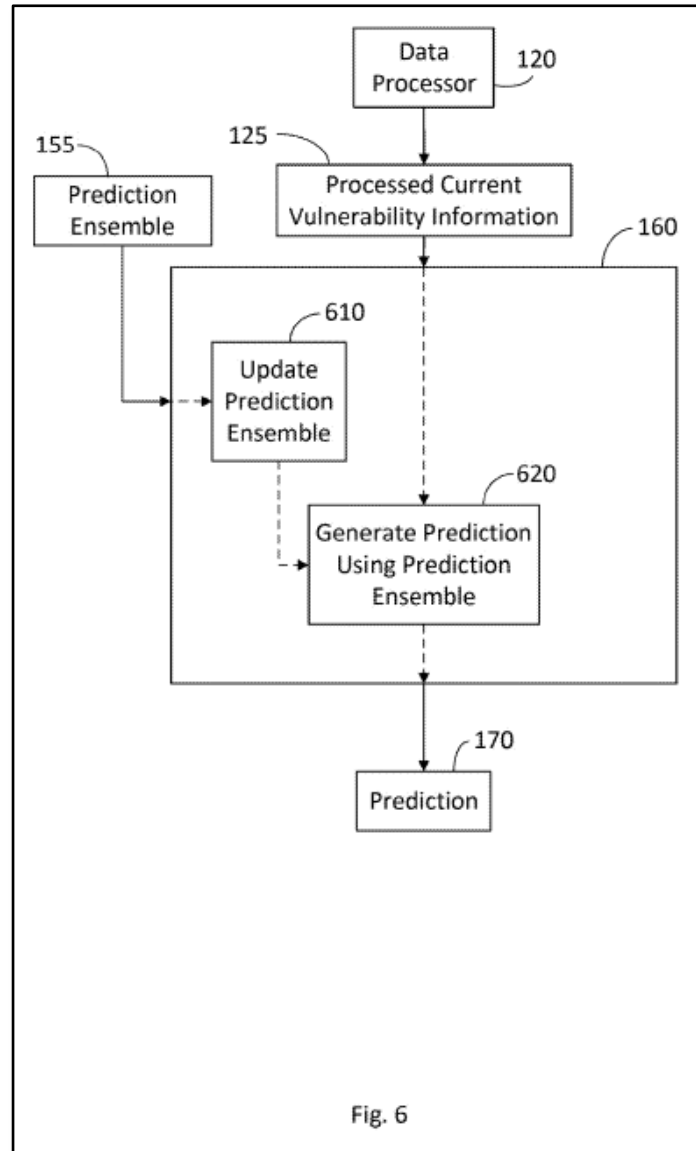


Smyth, Fig. 7; *supra*, claim 1. *Smyth* applies a prediction engine 160 (“learned function”) configured via machine learning algorithms 531-534 to vulnerability data 125 to estimate a likelihood of exploitation of the software vulnerability. *See generally*, Claim 1. *Smyth*’s prediction engine 160 is associated with a prediction ensemble 155 (“prediction model”) derived from machine learning algorithms 531-534 trained with data from the plurality of datasets, such as NVD, MITRE CVE List, VirusTotal Samples and Reports, Online Discussions, Proprietary Data Sources, and Other Data Sources (“*derived from at least one machine learning algorithms and a plurality of datasets associated with software vulnerabilities*”). *Id.*, Claim 1(b)-1(f). *Smyth-Phillips* further teaches accessing data from online discussions from the dark web, including discussion of vulnerabilities by users posting hacker-related content.

Id., Claim 1(b) (detailing motivation to combine). *Smyth*'s prediction engine 160 generates a prediction 170 that predicts an actual exploitation of the software vulnerability before disclosure based on at least hacker communications from the plurality of datasets. *Id.*, Claim 1(f)(1).

I. Claim 15 - The computing device of claim 14, wherein the prediction model is at least one classification model that outputs from features of the software vulnerability a score indicative of the likelihood of exploitation.

Smyth's prediction engine 160 includes prediction ensemble 155 ("prediction model") which is a classification model that applies one or more current vulnerability features to compute a prediction 170 defining the likelihood the software vulnerability will be exploited. This process is depicted below in Figure 6:



Smyth, Fig. 6. “This prediction 170 is a value that is within at least one of the thresholds previously defined by the computer administrator.” *Id.*, 15:8-10. Prediction value 170 is compared to thresholds to define a threat level and is therefore a “score indicative of the likelihood of exploitation” as claimed. *Dec.*, ¶136; *supra*, claim 2.

J. Claim 17 - The computing device of claim 14, wherein the computing device is further configured to apply the learned function to data associated with each of a plurality of new software vulnerabilities and compute a respective likelihood of exploitation for each of the plurality of new software vulnerabilities.

Smyth's prediction engine 160 applies its machine learning ensemble 155 (“learned function”) to online discussion data for vulnerabilities mentioned, but not yet exploited to compute a likelihood of exploitation. *Supra*, Claim 1(f), Claim 10. As discussed in Claim 1(f), *Smyth* teaches that “current vulnerability information 125 contains vulnerability information about a threat that is newly discovered” and includes “more current information such as online discussions 240[.]” *Supra*, Claim 1(f); *Smyth*, 11:1-8; *see also id.*, 14:55-57 (“The current vulnerability may be newly discovered, newly relied upon by malicious actors, or the like”). *Smyth* therefore applies its ensemble 155 to data from new software vulnerabilities discussed in online forums and computes the respective likelihood of exploitation of these new vulnerabilities. *Supra*, Claim 1(f)(1); *Dec.*, ¶137.

K. Claim 18

1. Claim 18[Pre]: “A method of prioritizing vulnerabilities using cyber threat intelligence, comprising:”

Smyth teaches methods for using a “predictive engine for analyzing existing vulnerability information to determine the likelihood of a vulnerability being exploited by malicious actors against a particular computer or network of computers.” *Smyth*, 1:7-11, 2:25-50, 3:63-67, 4:4-34, 11:22-25, claims 1 and 10.

Smyth explains “[t]his likelihood of exploitation information helps prioritize allocation of the limited available personnel and resources for securing a network or a system.” *Id.*, 2:19-21. *Smyth* thus discloses a method of prioritizing vulnerabilities using cyber threat intelligence. *Dec.*, ¶138.

2. ***Claim 18(a): “utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations including:”***

Smyth discloses a processor in operable communication with a main memory and secondary memory storing instructions that are executed by the processor for performing the claimed operations. *Supra*, Claim 1(a).

3. ***Claim 18(b): “accessing a plurality of datasets associated with a predetermined set of data sources, at least a portion of the plurality of datasets defining training data including hacker communications;”***

Smyth-Phillips teaches accessing a plurality of datasets from predetermined data sources (*see Smyth*, Fig. 4), where these datasets include online discussions 240 comprising hacker communications as modified by *Phillips* that is used to train *Smyth*’s prediction models. *Supra*, Claim 1(b).

4. ***Claim 18(c): “applying learning algorithms to the training data to generate a predictive model configured to predict a likelihood of exploitation of respective software vulnerabilities;”***

Smyth applies various learning algorithms (e.g., 531-534) to the training data to generate prediction ensemble 155 configured to predict a likelihood of exploitation of software vulnerabilities. *Supra*, Claim 1(d).

5. ***Claim 18(d): “accessing one or more features associated with a software vulnerability; and”***

Smyth teaches accessing features associated with a software vulnerability, including features from NVD, the MITRE CVE List, VirusTotal Samples and Reports, Online Discussions, Proprietary Data Sources, and Other Data Sources. *Supra*, Claim 1(e).

6. ***Claim 18(e): “computing, by applying the one or more features to the predictive model, one or more values defining a likelihood of exploitation associated with the software vulnerability, wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.”***

Smyth teaches a prediction engine 160 that applies one or more of the above current vulnerability features to prediction ensemble 155 to compute prediction 170. *Supra*, Claim 1(f) (noting *Smyth* computes the exploitability of a vulnerability defined by values such as exploited or not exploited that are defined by prediction 170 that indicates the likelihood of the prediction). *Smyth-Phillips* teaches prediction 170 predicts an actual exploitation of the software vulnerabilities before disclosure based on hacker communications from the training data. *Supra*, Claim 1(f)(1).

L. Claim 19 - The method of claim 18, further comprising generating a plurality of estimation outputs based on the one or more values to derive an overall quantitative score.

Smyth teaches its prediction ensemble generates a plurality of estimation outputs from multiple stacked algorithms and uses those outputs to derive an overall

prediction 170 that is compared to thresholds to define a threat level (“*overall quantitative score*”). *Supra*, Claim 2.

M. Claim 20 - The method of claim 18, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.

Smyth teaches accessing a plurality of datasets, including vulnerability data for publicly disclosed vulnerabilities (e.g., from the public National Vulnerability Database, MITRE CVE List, Other Data Sources, etc.) and exploits data that includes exploits used in real world attacks (e.g., from VirusTotal Samples and Reports and Online Discussions). *Supra*, Claim 3; *Smyth*, 4:35-55, Fig. 2.

VII. GROUND 3: CLAIMS 4 AND 16 ARE OBVIOUS OVER *SMYTH*, *PHILLIPS*, AND *NUNES*

A. Claim 4 - The method of claim 3, further comprising: aligning the exploits data with the vulnerability data; and cleaning the exploits data of noise and predetermined portions of the exploits data that is irrelevant to associated software vulnerabilities.

Smyth teaches the method of claim 3. *Supra*, Ground 1, Claim 3. *Smyth* further teaches “identifier[s]” for aligning exploits data with vulnerability data “so that interested individuals can associate the specific malicious software with the vulnerability being used by the specific malicious software.” *Smyth*, 5:4-9, 5:64-6:10. Data sources 210-260 “rely on a common identifier associating information with a particular vulnerability.” *Id.*, 4:52-65. *Smyth*’s predictive engine “relies on this common identifier to associate information about the vulnerability collected

from multiple disparate sources 110 in a manner that is more easily considered by the machine learning, statistical, and other predictive models used.” *Id.*, 5:32-39.

While *Smyth-Phillips* teaches accessing exploits data, including from online discussions in dark web forums, and aligning that exploits data with vulnerability data for analysis by *Smyth*’s predictive models, *Smyth-Phillips* does not expressly describe further processing that exploits data to remove noise and irrelevant data. The inventors of the ’897 Patent dedicated this concept to the public, however, in 2016 when they published *Nunes*. Specifically, *Nunes* teaches a “parser” that “extract[s] specific information from...hacker forums (discussion regarding services and threats).” *Nunes*, 2. To detect relevant topics only and therefore reduce noise and filter out irrelevant data, classifiers are integrated into *Nunes*’ parser. *Id.*, 3 (“classifiers are integrated into the parser to filter out products and topics relating to drugs, weapons, etc. not relevant to malicious hacking”); *see also id.*, 2 (implementing “learning models to separate relevant information from noise in the data collected from these online platforms”).

A POSITA would have been motivated to implement *Nunes*’ parser with an integrated classifier in *Smyth*’s system for at least the benefits expressed in *Nunes*. *Dec.*, ¶¶147-149. *Nunes* acknowledges that “only a fraction of [online discussion] topics with posts on these forums contain information that is relevant to malicious hacking” and thus uses its parser equipped with a classifier “to identify relevant

topics.” *Nunes*, 4. Because this data is “noisy” and “not suitable to use directly as input to a learning model[,]” *Nunes* cleans this data of noise and irrelevant data. *Id.*, 3. Focusing *Smyth*’s learning models only on relevant hacker data from forums and removing predetermined irrelevant data (e.g., discussions on drugs, weapons, etc.) would improve classification and reduce the computational load of *Smyth*’s learning models. *Dec.*, ¶149.

A POSITA would have appreciated a reasonable expectation of success modifying *Smyth* to implement *Nunes*’ parser with an integrated classifier. *Dec.*, ¶150. *Smyth* describes several known techniques to collect and process data for its predictive models. *Smyth*, 5:49-52 (collecting data by, e.g., “web page scraping” or “other known techniques”). *Nunes*’ parser was a well-known technique to collect online forum data, including dark web data, and would have integrated seamlessly into *Smyth*’s system to collect forum data 240 for *Smyth*’s predictive models. *Dec.*, ¶150. Further, *Smyth* describes example processing performed on this collected data. *Smyth*, 6:22-55 (describing indexing and transforming data to “facilitate[] processing by the predictive models”). *Smyth* notes embodiments may “employ other techniques not specifically described here...and provide [the data] in a manner suitable for the prediction engine 160.” *Id.* In embodiments, *Smyth* can vary processing based on the data source. *Id.*, 9:6-17. Thus, it would have been well within the skill of a POSITA to implement *Nunes*’ classifier integrated in its parser

to process the collected data and filter out noise and predetermined irrelevant data.

Dec., ¶¶149-150.

B. Claim 16 - The computing device of claim 14, wherein the predictive model includes a Random Forest (RF) method including multiple decision tree predictors applied in combination to classify the software vulnerability which is used to estimate the likelihood of exploitation.

Smyth-Phillips teaches the computing device of claim 14. *Supra*, Ground 1, Claim 14. *Smyth* further teaches its prediction engine 160 (“predictive model”) comprises an ensemble of different machine learning algorithms, including “tree-based model” 531. *Smyth*, 9:20-45, Fig. 5. “The prediction engine 160 takes [] processed current vulnerability information 125 and uses the prediction ensemble 155...to generate a prediction 170 of the likelihood the current vulnerability may be employed[.]” *Id.*, 4:24-29, 10:61-64, 11:18-25. While *Smyth* discloses using tree-based models to classify the software vulnerability, *Smyth* does not expressly state it uses a Random Forest tree-based model. *Nunes* discloses “well-known classification techniques” including “random forest (RF)[.]” *Nunes*, 2.

It would have been obvious to apply *Nunes*’ well-known RF classification technique as the specific tree-based model in *Smyth* because this is a simple substitution of one known element for another to obtain predictable results. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-421 (2007); *Dec.*, ¶¶151-152. Applying an RF model in *Smyth*’s predictive engine would improve classification. *Dec.*, ¶152.

RF models aggregate predictions with high accuracy, are robust to noise, and are non-parametric such that these models may be applied to a variety of datasets, making them suitable for *Smyth*'s invention. *Dec.*, ¶152. A POSITA would have appreciated a reasonable expectation of success implementing an RF model as one of the predictive algorithms in *Smyth* because this was a well-known classification technique and sub-species of the tree-based models already used in *Smyth*. *Dec.*, ¶152.

VIII. GROUND 4: CLAIM 6 IS OBVIOUS OVER *SMYTH*, *PHILLIPS*, AND *KUPERMAN*

A. Claim 6 - The method of claim 1, further comprising testing the classification models by applying additional training data and one or more algorithms and evaluation metrics to optimize the classification models until the classification models compute the likelihood of exploitation according to a predefined error rate.

Smyth teaches a process for testing its classification models. *Smyth*, 10:5-41. Specifically, combined data set 123 is split into training 512 and testing sets 514. *Id.*, 10:13-15. If training set 512 is imbalanced, *Smyth* applies additional training data to rebalance the set “using techniques such as resampling, over sampling, or under sampling, synthetic minority over-sampling, and random minority over-sampling.” *Id.*, 10:15-27. Rebalanced training data 512 is then used to train the predictive models. *Id.* Testing sets 514 are then used to test the retrained models. *Id.*, 10:28-41. To optimize these models, *Smyth* applies a “modeling scheme 130” that employs various algorithms that overlap with the optimization algorithms disclosed

in the '897 Patent's specification. *Compare Smyth*, 10:55-59 (describing “majority voting” and “boosting”) with '897 Patent, Fig. 13 (205 depicting testing and improving classification models using algorithms like “Boosting” and “Majority voting”).

Smyth's modeling scheme 130 may also include “adjustments or biases for consideration when retraining” to “tailor the predictions 170 in a manner that provides a more useful prediction 170.” *Smyth*, 9:50-54. Like the claimed invention, *Smyth* adjusts data based on evaluation metrics to optimize the model “so that, for example, a higher true positive rate may be achieved while reducing the rate of false positives.” *Compare Smyth*, 9:60-64 with '897 Patent, 15:11-30 (Table 4 listing evaluation metrics such as “TPR” true positive rate and “FPR” false positive rate).

While *Smyth* discloses using additional training data, algorithms, and evaluation metrics to optimize the classification models to verify their performance, *Smyth* does not expressly describe optimizing the classification models “according to a predefined error rate.” *Kuperman* teaches an optimization process for a malware classification model, where a minimum desired performance level is evaluated based on accuracy metrics such as “False Positive %” and “False Negative %” such that a training error rate determines whether the model exceeds a predefined “desired performance level” and can be marked “live” for use. *Kuperman*, 19:61-20:22, Fig. 6 (depicting error rate above predefined desired performance level).

A POSITA would have been motivated to modify *Smyth* to optimize its models according to a predefined error rate, as taught by *Kuperman*, to provide a known and reliable metric to verify model performance. *Dec.*, ¶¶156-159. *Smyth* already discloses testing its models “against the testing set 514 so that their performance can be verified.” *Smyth*, 10:44-47. Verifying performance according to a predefined error rate provides a simple mechanism to interpret results and provides a good indicator for model accuracy. *Dec.*, ¶159. A POSITA would have appreciated a reasonable expectation of success making this modification because *Smyth* already describes optimizing and verifying model performance and a predefined error rate was a widely used metric for model optimization and verification that would have been suitable for *Smyth*’s methods. *Dec.*, ¶160.

IX. GROUND 5: CLAIMS 7 AND 13 ARE OBVIOUS OVER *SMYTH*, *PHILLIPS*, AND *MARIN*

A. Claim 7 - The method of claim 1, further comprising vectorizing text features derived from the plurality of datasets using term frequency-inverse document frequency to create a vocabulary of associated words.

Smyth teaches vectorizing text features derived from the plurality of datasets, including forum discussions, using “unstructured text analysis 430.” *Smyth*, 7:22-47 (“unstructured text analysis 430 is performed on certain portions of the vulnerability information so that features may be derived” such as “online discussions 240”), Fig. 4. *Smyth* further teaches in embodiments, “the unstructured text may be processed by generative statistical models[.]” *Id.*, 7:35-47 (listing example statistical models

such as latent Dirichlet allocation (LDA) and sentiment analysis). While *Smyth* teaches or suggests using certain text vectorization techniques, *Smyth* does not expressly discuss using “term frequency-inverse document frequency” (TF-IDF) to vectorize text features.

TF-IDF was a known text vectorization technique taught by *Marin*, a paper co-authored by a co-inventor of the '897 Patent. *Marin*, 1. *Marin* teaches analyzing websites on the darkweb for malicious hacking products and using machine learning to categorize and study those products. *Marin*, 1. *Marin* teaches vectorizing text features for its machine learning models, such as title, description, posting date, product name, using “standard term frequency – inverse document frequency (TF-IDF)[.]” *Id.*, 2; *see also id.*, 3 (Table II showing “performance of each TF-IDF vectorization”). It would have been obvious to vectorize text information from *Smyth-Phillip*’s online forums using TF-IDF in conjunction with LDA to create a vocabulary of associated words. *Dec.*, ¶¶163-164. TF-IDF is a simple and computationally inexpensive vectorization technique that accurately determines the relevance of words within text. *Dec.*, ¶164. LDA relies on topic modeling. *Dec.*, ¶164. A POSITA would have been motivated to implement TF-IDF and LDA together to enhance classification by correlating frequent keywords (TF-IDF) with latent topics and themes (LDA). *Dec.*, ¶164. A POSITA would have had a reasonable expectation of success making this simple modification because *Smyth* already uses

text vectorization and TF-IDF was a widely known and commonly used technique for the same with understood benefits. *Dec.*, ¶165.

B. Claim 13 - The method of claim 1, further comprising vectorizing text features derived from textual content of the plurality of datasets using a predetermined natural language process (NPL).

Smyth-Phillips-Marin teaches vectorizing text features from textual content of a plurality of datasets using TF-IDF plus LDA. *Supra*, Claim 7. TF-IDF/LDA is a predetermined natural language process. *Dec.*, ¶166.

X. GROUND 6: CLAIMS 8 AND 12 ARE OBVIOUS OVER *SMYTH*, *PHILLIPS*, AND *BOZORGI*

A. Claim 8 - The method of claim 1, further comprising: sorting vulnerabilities associated with the plurality of datasets according to time; training the classification model using the training data, the training data defining a first subset of the plurality of datasets associated with a predetermined period of time; and testing the classification model using a second subset of the plurality of datasets associated with the predetermined period of time.

Smyth teaches training its classification model with training data from the plurality of datasets in combined data set 123. *Supra*, Claim 1(b), 1(d). *Smyth* splits combined data set 123 into training and testing sets. *Supra*, Claim 6. “One such split of the combined data set 123 may separate information for which impacts are known from information for which impacts are not quantified.” *Smyth*, 10:9-12. For example, “testing set 514 may be formed of information for which the impacts of certain vulnerabilities are known, so an actual threat level is derived from such impacts.” *Id.*, 10:29-32. *Smyth* therefore teaches sorting and dividing vulnerability

data for training and testing based on whether the impact is known or not. *Dec.*, ¶170. *Smyth*, however, does not sort and divide vulnerability data for training and testing based on time.

Bozorgi teaches a vulnerability prioritization scheme that predicts “whether and how soon a vulnerability is likely to be exploited.” *Bozorgi*, 1. Like *Smyth*, *Bozorgi* trains a classifier “on known vulnerabilities to make predictions about new ones.” *Id.*, 5. To accomplish this, *Bozorgi* trains and initializes its classifier with all vulnerabilities “reported between time 0 and time t that have known outcomes” and then further tests this classifier with “vulnerabilities that appear in the next time interval” between $t+1$ and $t+T$ based on new vulnerabilities arriving between $t+1$ and $t+T$, and then “update[s] the classifier with known vulnerability outcomes.” *Id.*, 6. *Bozorgi*, therefore, trains its classifier with a first subset of data “associated with a predetermined period of time” and tests its classifier with a second subset within that predetermined period of time. *Dec.*, ¶¶171-172. A POSITA would have understood that the “time 0 and time t ” plus the “next time interval” (i.e., time $t+1$ to $t+T$) would constitute a “predetermined time period of time” (i.e., time 0 to $t+T$) as claimed. *Id.*, ¶172. A POSITA would have thus understood that *Bozorgi* sorts its vulnerability data based on time to train and test data within a predetermined period of time. *Dec.*, ¶¶171-172. It would also have been obvious to train and test over the same time

interval using different subsets of data to evaluate the classifier's performance. *Dec.*, ¶175.

A POSITA would have been motivated to modify *Smyth* with *Bozorgi* to sort, train, and test *Smyth*'s vulnerability data according to time because this would be a natural splitting of the data, based on time ordering (e.g., past and future) would better reflect real-world applications. *Dec.*, ¶¶173-175. A POSITA would have appreciated a reasonable expectation of success making this minor modification. *Dec.*, ¶174. *Smyth* teaches its data sources 110 can additionally include the "date a vulnerability was released" and has a similar process for splitting training and test data. *Smyth*, 7:9-21, 10:5-12. Thus, it would have required simple programming changes to sort vulnerability data by dates/times and split the training and test data according to the same. *Dec.*, ¶¶174.

B. Claim 12 - The method of claim 1, further comprising: accessing one or more features indicative of temporal connections between at least two of: a time associated with discussion of a vulnerability by users posting hacking-related content at a web forum prior to disclosure to a public vulnerability database; a time associated with disclosure of the vulnerability to the public vulnerability database; and a time associated with exploitation of the vulnerability as obtained through exploits data associated with real-world exploitation of the vulnerability.

Smyth-Phillips teaches deriving features from various data sources 100, including features from web forums containing hacking content. *Supra*, Claim 1(c). For example, *Smyth* teaches using "unstructured text analysis 430" on portions of the vulnerability information such as online discussions so that features may be

derived for *Smyth*'s predictive models. *Id.*; *Smyth*, 7:22-35. *Phillips* also derives features from metadata of hacker forums. *Supra*, Claim 1(c); *Phillips*, 1-2 (“Methodology”). This includes, or would have been obvious to include, “timestamps” for forum posts. *Phillips*, 1. *Dec.*, ¶176. *Smyth-Phillips* therefore teaches or suggests accessing features indicative of “a time associated with discussion of a vulnerability by users posting hacking-related content at a web forum prior to disclosure to a public vulnerability database[.]” *Dec.*, ¶176.

While *Smyth* teaches deriving features from known vulnerability data, including CVE entries, and notes CVE entries include “the date the entry was created[.]” *Smyth* does not expressly state it derives CVE date information for its classification process. *Bozorgi* teaches extracting CVE time stamps “as features for classification[.]” *Bozorgi*, 3. Table 2 provides a list of features extracted for each vulnerability *Bozorgi* analyzes, including “CVE Dates.” *Id.*, 4 (describing “integer-valued features” that “encode useful information, such as the date when a vulnerability was first disclosed”). *Bozorgi* therefore teaches accessing features indicative of a “time associated with disclosure of the vulnerability to the public vulnerability database[.]” *Dec.*, ¶177. A POSITA would have been motivated and would have appreciated a reasonable expectation of success modifying *Smyth-Phillips* pursuant to *Bozorgi* to access temporal features for the same benefits

discussed above in claim 8 regarding improving classification with simple programming changes. *Supra*, Claim 8; *Dec.*, ¶¶176-177.

XI. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests *inter partes* review of the Challenged Claims.

Respectfully submitted,

By: /s/ Paul R. Hart

Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER

XII. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)

A. Real Party-In-Interest

Petitioner is the real parties-in-interest. 37 C.F.R. § 42.8(b)(1).

B. Related Matters

The '897 Patent is presently the subject of a patent infringement lawsuit filed against Petitioner in the United States District Court for the Western District of Texas, *Skysong Innovations, LLC v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.*, Case No. 7:25-cv-00040-DC-DTG. The '897 Patent has also been asserted against Fortinet, Inc. in the United States District Court for the Eastern District of Texas, *Skysong Innovations, LLC v. Fortinet, Inc.*, Case No. 2:25-cv-00098.

C. Lead and Back-Up Counsel

Petitioners provides the following designation and service information for lead and back-up counsel. 37 C.F.R. § 42.8(b)(3) and (b)(4).

Lead Counsel	Back-Up Counsel
Paul R. Hart, (Reg. No. 59,646) Paul.Hart@eriseip.com <u>PTAB@eriseip.com</u>	Adam P. Seitz (Reg. No. 52,206) Adam.Seitz@eriseip.com
<u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 717 17 th St., Suite 1400 Denver, Colorado 80202 Telephone: (913) 777-5600 Fax: (913) 777-5601	<u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601
	Hunter A. Horton (<i>pro hac vice</i> to be submitted) hunter.horton@eriseip.com

	<u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601 Nathan S. Byers (Reg. No. 81,095) <u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 717 17 th St., Suite 1400 Denver, Colorado 80202 Telephone: (913) 777-5600 Fax: (913) 777-5601
--	---

D. 37 C.F.R. § 42.8(b)(4) – Service Information

Please address all correspondence to the lead and back-up counsel as shown above. Petitioner consents to electronic service by e-mail at the e-mail addresses provided above.

CLAIM LISTING APPENDIX

Claim Designation	Claim Language
Claim 1(Pre)	A method for assessing a likelihood of exploitation of software vulnerabilities, comprising:
Claim 1(a)	utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations, including:
Claim 1(b)	accessing a plurality of datasets associated with a predetermined set of data sources, the plurality of datasets including training data comprising hacker communications;
Claim 1(c)	accessing features from the plurality of datasets that include measures computed from social connections of users posting hacking-related content;

Claim Designation	Claim Language
Claim 1(d)	applying learning algorithms to the training data to generate classification models that are configured to predict class labels defining a likelihood of exploitation of respective software vulnerabilities;
Claim 1(e)	accessing one or more features associated with a software vulnerability; and
Claim 1(f)	computing, by applying the one or more features to the classification model, a class label defining one or more values defining a likelihood of exploitation associated with the software vulnerability,
Claim 1(f)(1)	wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.
Claim 2	The method of claim 1, further comprising generating a plurality of estimation outputs based on the one or more values to derive an overall quantitative score.
Claim 3	The method of claim 1, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.
Claim 4	The method of claim 3, further comprising: aligning the exploits data with the vulnerability data; and cleaning the exploits data of noise and predetermined portions of the exploits data that is irrelevant to associated software vulnerabilities.
Claim 5	The method of claim 1, wherein certain features correspond to a known vulnerability obtained from the plurality of datasets.
Claim 6	The method of claim 1, further comprising testing the classification models by applying additional training data and one or more algorithms and evaluation metrics to optimize the classification models until the classification models compute the likelihood of exploitation according to a predefined error rate.
Claim 7	The method of claim 1, further comprising vectorizing text features derived from the plurality of datasets using term frequency-inverse document frequency to create a vocabulary of associated words.

Claim Designation	Claim Language
Claim 8	The method of claim 1, further comprising: sorting vulnerabilities associated with the plurality of datasets according to time; training the classification model using the training data, the training data defining a first subset of the plurality of datasets associated with a predetermined period of time; and testing the classification model using a second subset of the plurality of datasets associated with the predetermined period of time.
Claim 9	The method of claim 1, further comprising computing mutual information from the plurality of datasets informative as to what information a given feature provides about another feature.
Claim 10	The method of claim 1, further comprising: detecting, from the plurality of datasets, vulnerabilities that appear before an associated exploitation date.
Claim 11	The method of claim 1, further comprising: accessing features from the plurality of datasets that measure a centrality of the users in a social graph.
Claim 12	The method of claim 1, further comprising: accessing one or more features indicative of temporal connections between at least two of: a time associated with discussion of a vulnerability by users posting hacking-related content at a web forum prior to disclosure to a public vulnerability database; a time associated with disclosure of the vulnerability to the public vulnerability database; and a time associated with exploitation of the vulnerability as obtained through exploits data associated with real-world exploitation of the vulnerability.
Claim 13	The method of claim 1, further comprising vectorizing text features derived from textual content of the plurality of datasets using a predetermined natural language process (NLP).

Claim Designation	Claim Language
Claim 14	A computing device, configured via machine learning to apply a learned function to data associated with a software vulnerability to estimate a likelihood of exploitation of the software vulnerability, the learned function associated with a prediction model derived from at least one machine learning algorithm and a plurality of datasets associated with software vulnerabilities, the plurality of datasets including information associated with discussion of vulnerabilities by users posting hacking-related content, wherein the likelihood of exploitation predicts an actual exploitation of the software vulnerability before disclosure based on hacker communications from the plurality of datasets.
Claim 15	The computing device of claim 14, wherein the prediction model is at least one classification model that outputs from features of the software vulnerability a score indicative of the likelihood of exploitation.
Claim 16	The computing device of claim 14, wherein the predictive model includes a Random Forest (RF) method including multiple decision tree predictors applied in combination to classify the software vulnerability which is used to estimate the likelihood of exploitation.
Claim 17	The computing device of claim 14, wherein the computing device is further configured to apply the learned function to data associated with each of a plurality of new software vulnerabilities and compute a respective likelihood of exploitation for each of the plurality of new software vulnerabilities.
Claim 18(Pre)	A method of prioritizing vulnerabilities using cyber threat intelligence, comprising:
Claim 18(a)	utilizing a processor in operable communication with at least one memory for storing instructions that are executed by the processor to perform operations including:
Claim 18(b)	accessing a plurality of datasets associated with a predetermined set of data sources, at least a portion of the plurality of datasets defining training data including hacker communications;

Claim Designation	Claim Language
Claim 18(c)	applying learning algorithms to the training data to generate a predictive model configured to predict a likelihood of exploitation of respective software vulnerabilities;
Claim 18(d)	accessing one or more features associated with a software vulnerability; and
Claim 18(e)	computing, by applying the one or more features to the predictive model, one or more values defining a likelihood of exploitation associated with the software vulnerability, wherein the likelihood of exploitation predicts an actual exploitation of the respective software vulnerabilities before disclosure based on the hacker communications from the training data.
Claim 19	The method of claim 18, further comprising generating a plurality of estimation outputs based on the one or more values to derive an overall quantitative score.
Claim 20	The method of claim 18, wherein the plurality of datasets include vulnerability data for vulnerabilities that are publicly disclosed and obtaining exploits data for exploits that were used in real world attacks.

APPENDIX OF EXHIBITS

Exhibit 1001	U.S. Patent No. 11,892,897 (“’897 Patent”)
Exhibit 1002	File History of the ’897 Patent (“’897 File History”)
Exhibit 1003	Declaration of Dr. Nicholas Bambos (“Dec.”)
Exhibit 1004	Carl Sabottke et. al., <u>Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits</u> , Proceedings of the 24 th USENIX Security Symposium (2015) (“Sabottke”)
Exhibit 1005	U.S. Patent No. 10,264,009 to Smyth et al. (“Smyth”)
Exhibit 1006	Elizabeth Phillips et al., <u>Extracting Social Structure from DarkWeb Forums</u> , SOTICS 2015: The Fifth International Conference on Social Media Technologies, Communication, and Informatics, (2015) (“Phillips”)
Exhibit 1007	Eric Nunes et al., <u>Darknet and Deepnet Mining for Proactive Cybersecurity Threat Intelligence</u> , Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (“Nunes”)
Exhibit 1008	U.S. Patent No. 10,652,254 to Kuperman et al. (“Kuperman”)
Exhibit 1009	Ericsson Marin et al., <u>Product Offerings in Malicious Hacker Markets</u> , Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (“Marin”)
Exhibit 1010	Mehran Bozorgi et al., <u>Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits</u> , Proceedings of the Sixteenth ACM Conference on Knowledge Discovery and Data Mining (KDD 2010) (“Bozorgi”)
Exhibit 1011	<i>Intentionally left blank</i>
Exhibit 1012	U.S. Patent No. 9,294,498 to Yampolskiy et al. (“Yampolskiy”)
Exhibit 1013	Victor Benjamin et al., <u>Securing Cyberspace: Identifying Key Actors in Hacker Communities</u> , Proceedings of the 2012 IEEE International Conference on Intelligence and Security Informatics (2012) (“Benjamin”)
Exhibit 1014	Declaration of June Munford (“Munford Dec.”)
Exhibit 1015	Mitch Macdonald et al., <u>Identifying Digital Threats in a Hacker Web Forum</u> , IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, (2015) (“Macdonald”)
Exhibit 1016	<u>CTC Sentinel</u> , Volume 6, Issue 7, July 2013. Combating Terrorism Center at West Point. (“CTC Sentinel”)
Exhibit 1017	Vivian Fiona Guetler, <u>Exploring Cyberterrorism, Topic Models and Social Networks off Jihadists Dark Web Forums: A</u>

	<u>Computational Social Science Approach</u> , Graduate Theses, Dissertations, and Problem Reports (2022) (“ <i>Guetler</i> ”)
Exhibit 1018	Screenshot of VirusTotal, found at https://www.virustotal.com/getstarted/vulnerability-management , captured June 12, 2025 (“ <i>VirusTotal.com Screenshot</i> ”)
Exhibit 1019	Elizabeth Phillips et al., <u>Applying Social Network Analysis to Security</u> , International Conference on Cyber Security for Sustainable Society (2015)
Exhibit 1020	<i>Intentionally left blank</i>
Exhibit 1021	Barreras et al., <u>Improving Transparency of the Colombian Peace Treaty with NLP</u> (“ <i>Barreras</i> ”)
Exhibit 1022	Tianjun Fu et al., <u>A Focused Crawler for Dark Web Forums</u> , Journal of the American Society for Information Science and Technology (2010) (“ <i>Fu</i> ”)
Exhibit 1023	Victor Benjamin et al., <u>Exploring Threats and Vulnerabilities in Hacker Web: Forums, IRC and Carding Shops</u> , IEEE Intelligence and Security Informatics (2015)
Exhibit 1024	U.S. Patent No. 11,031,135 to Gates et. al (“ <i>Gates</i> ”)
Exhibit 1025	Mohammed Almukaynizi et al., <u>Predicting Cyber Threats through Hacker Social Networks in Darkweb and Deepweb Forums</u> , CSSSA’s Annual Conference on Computational Social Science (2017), (“ <i>Almukaynizi</i> ”)
Exhibit 1026	Thomas J. Holt et al., <u>Examining the Social Networks of Malware Writers and Hackers</u> , International Journal of Cyber Criminology (2012) (“ <i>Holt</i> ”)
Exhibit 1027	Hoa Khanh Dam et al., <u>Automatic feature learning for predicting vulnerable software components</u> , IEEE Transactions of Software Engineering, (2018) (“ <i>Dam</i> ”)
Exhibit 1028	Michel Edkrantz, <u>Predicting Exploit Likelihood for Cyber Vulnerabilities with Machine Learning</u> , Chalmers Univ. of Tech., (2015) (“ <i>Edkrantz</i> ”)

CERTIFICATION OF WORD COUNT

The undersigned certifies pursuant to 37 C.F.R. §42.24 that the foregoing Petition for *Inter Partes* Review, excluding any table of contents, mandatory notices under 37 C.F.R. §42.8, certificates of service or word count, or appendix of exhibits, contains 13,915 words according to the word-processing program used to prepare this document (Microsoft Word).

Dated: July 25, 2025

Respectfully submitted,

By: /s/ Paul R. Hart
Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER

**CERTIFICATE OF SERVICE ON PATENT OWNER
UNDER 37 C.F.R. § 42.105**

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b), the undersigned certifies that on July 25, 2025, a complete and entire copy of this Petition for *Inter Partes* Review and Exhibits were provided via Federal Express to the Patent Owner by serving the correspondence address of record for the '897 Patent:

POLSINELLI PC
PO Box 140310
Kansas City, MO 64114-0310

Further, a courtesy copy of this Petition for *Inter Partes* Review was sent via email to Patent Owner's litigation counsel:

Jay P. Kesan (jay.kesan@kkd-law.com)
Cecil E. Key (cecil.key@kkd-law.com)
Alfonso Garcia Chan (achan@cahill.com)
Britton F. Davis (bdavis@cahill.com)
Christopher C. Campbell (ccampbell@cahill.com)
Halima Shukri Ndai (hndai@cahill.com)

Respectfully submitted,

By: /s/ Paul R. Hart
Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER