

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TREND MICRO, INC.
Petitioner

v.

WEBROOT INC.

Case No. IPR2023-01380
Patent No. 11,409,869

**PETITION FOR INTER PARTES REVIEW
OF U.S. PATENT NO. 11,409,869**

Filed on behalf of Trend Micro, Inc.
By: Robert M. Hansen
Matthew J. Anderson
Scott B. Amankwatia
The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191

Tel: (703) 391-2900
Fax: (703) 391-2901

TABLE OF CONTENTS

I. Mandatory Notices (37 C.F.R. § 42.8(a)(1))	1
II. Payment of Fees.....	4
III. Grounds for Standing (37 C.F.R. § 42.104(a)).....	4
IV. Identification of Challenge (37 C.F.R. § 42.104(b))	4
A. Citation of Prior Art	4
B. Statutory Grounds for Challenge	5
V. The '869 Patent.....	6
A. Filing Date of the '869 patent	6
B. Specification.....	6
C. Summary of the Prosecution History	7
D. Level of Ordinary Skill in the Art.....	9
E. Claim Construction.....	9
VI. Grounds of Rejection.....	10
A. Ground 1: Claims 1-4, 8, 10-25, 29, 32-39, 43, 46-53, and 57 are unpatentable under 35 U.S.C. § 103 over Tahan in view of Menahem and Forman	11
1. Claim 1.....	15
a. A computer-implemented method comprising:	15
b. identifying, by a knowledge module, static data points that may be indicative of either a harmful or benign executable file;.....	15
c. associating, by the knowledge module, the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;	16
d. identifying an executable file to be evaluated;	17

e. extracting, by the knowledge module, a plurality of static data points from the identified executable file;.....	18
f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files, ...	19
g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and....	23
h. providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files.	28
2. Claim 2.....	31
The computer-implemented method according to claim 1, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.	31
3. Claim 3.....	32
The computer-implemented method according to claim 1, wherein the one or more support vector machines builds the probabilistic model by performing data analysis and pattern recognition on the one or more feature vectors.....	32
4. Claim 4.....	33
The computer-implemented method according to claim 1, wherein the probabilistic model indicates whether the executable file is harmful.	33
5. Claim 8.....	34

The computer-implemented method according to claim 1, wherein the plurality of static data points represent predefined character strings in the executable file.	34
6. Claim 10.....	35
a. A system comprising: at least one memory; and at least one processor operatively connected with the memory and configured to perform operation of:	35
b. identifying static data points that may be indicative of either a harmful or benign executable file;.....	35
c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;.....	36
d. identifying an executable file to be evaluated;	36
e. extracting a plurality of static data points from the identified executable file; and	36
f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files, ...	36
g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and....	37
h. providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files.	37
7. Claim 11.....	37

The system according to claim 10, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.	37
8. Claim 12.....	38
The system according to claim 10, wherein the one or more support vector machines builds the probabilistic model by performing data analysis and pattern recognition on the one or more feature vectors.	38
9. Claim 13.....	38
The system according to claim 10, wherein the probabilistic model indicates whether the executable file is harmful.	38
10. Claim 14.....	39
The system according to claim 10, wherein the plurality of static data points represent predefined character strings in the executable file.....	39
11. Claim 15.....	39
a. A computer-readable storage device containing instructions, that when executed on at least one processor, causing the processor to execute a process comprising:	39
b. static data points that may be indicative of either a harmful or benign executable file;	39
c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;	40
d. identifying an executable file to be evaluated;	40
e. extracting a plurality of static data points from the identified executable file;	40

f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files, ...	40
g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and....	40
h. providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files.	41
12. Claim 16.....	41
The computer-readable storage device according to claim 15, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.	41
13. Claim 17.....	41
The computer-readable storage device according to claim 15, wherein the plurality of static data points represent predefined character strings in the executable file.	41
14. Claim 18.....	42
a. A computer-implemented method comprising:	42
b. identifying static data points that may be indicative of either a harmful or benign executable file;.....	42
c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;.....	42
d. identifying an executable file to be evaluated;	42

e. extracting a plurality of static data points from the executable file;	42
f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files, ...	43
g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and....	43
h. evaluating the feature vector using a machine learning model to determine whether the executable file fits into one of the categories of files.....	43
15. Claim 19.....	44
The computer-implemented method according to claim 18, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.	
16. Claim 20.....	44
The computer-implemented method according to claim 18, wherein the machine learning model comprises an artificial neural network.....	
17. Claim 21.....	45
The computer-implemented method according to claim 18, wherein the machine learning model comprises a support vector machine.....	
18. Claim 22.....	45
The computer-implemented method according to claim 18, wherein the machine learning model comprises a machine learning decision tree.	
19. Claim 23.....	46

The computer-implemented method according to claim 18, wherein the machine learning model comprises a Bayesian network.....	46
20. Claim 24.....	46
The computer-implemented method according to claim 18, wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.....	46
21. Claim 25.....	47
The computer-implemented method according to claim 18, wherein the machine learning model comprises a clustering model.....	47
22. Claim 29.....	48
The computer-implemented method according to claim 26, wherein the plurality of static data points represent predefined character strings in the executable file.	48
23. Claim 32.....	48
a. A computer-readable storage device containing instructions, that when executed on a processor, cause the processor to execute a process comprising:	48
b. identifying static data points that may be indicative of either a harmful or benign executable file;.....	48
c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;.....	48
d. identifying an executable file to be evaluated;	49
e. extracting a plurality of static data points from the executable file;	49
f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data,	

the training data comprising files known to fit into one of the plurality of categories of files, ...	49
g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and....	49
h. evaluating the feature vector using a machine learning model to determine whether the executable file fits into one of the categories of files.....	49
24. Claim 33.....	50
The computer-readable storage device according to claim 32, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.	50
25. Claim 34.....	50
The computer-readable storage device according to claim 32, wherein the machine learning model comprises an artificial neural network.....	50
26. Claim 35.....	51
The computer-readable storage device according to claim 32, wherein the machine learning model comprises a support vector machine.....	51
27. Claim 36.....	51
The computer-readable storage device according to claim 32, wherein the machine learning model comprises a machine learning decision tree.	51
28. Claim 37.....	51
The computer-readable storage device according to claim 32, wherein the machine learning model comprises a Bayesian network.....	51
29. Claim 38.....	52

The computer-readable storage device according to claim 32, wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.....	52
30. Claim 39.....	52
The computer-readable storage device according to claim 32, wherein the machine learning model comprises a clustering model.....	52
31. Claim 43.....	53
The computer-readable storage device according to claim 32, wherein the plurality of static data points represent predefined character strings in the executable file.	53
32. Claim 46.....	53
a. A system comprising: at least one memory; at least one processor configured to perform operation of:	53
b. identifying static data points that may be indicative of either a harmful or benign executable file;.....	53
c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;.....	53
d. at least one processor configured to perform operation of:	54
e. identifying an executable file to be evaluated;	54
f. extracting a plurality of static data points from the executable file;.....	54
g. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files, ...	54

h. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and....	54
i. evaluating the feature vector using a machine learning model to determine whether the executable file fits into one of the categories of files.....	55
33. Claim 47.....	55
The system according to claim 46, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.	55
34. Claim 48.....	55
The system according to claim 46, wherein the machine learning model comprises an artificial neural network.....	55
35. Claim 49.....	56
The system according to claim 46, wherein the machine learning model comprises a support vector machine.....	56
36. Claim 50.....	56
The system according to claim 46, wherein the machine learning model comprises a machine learning decision tree.	56
37. Claim 51.....	57
The system according to claim 46, wherein the machine learning model comprises a Bayesian network.....	57
38. Claim 52.....	57
The system according to claim 46, wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.....	57

39. Claim 53.....	57
The system according to claim 46, wherein the machine learning model comprises a clustering model.....	57
40. Claim 57.....	58
The system according to claim 46, wherein the plurality of static data points represent predefined character strings in the executable file.....	58
B. Ground 2: Claims 5-7, 9, 26-28, 30-31, 40-42, 44-45, 54-56, and 58-59 are unpatentable under 35 U.S.C. § 103 over Tahan in view of Menahem, Forman, and Schultz.	58
1. Claim 5.....	60
The computer-implemented method according to claim 1, wherein the executable file is identified in response to a detected condition.	60
2. Claim 6.....	60
The computer-implemented method according to claim 5, wherein the detected condition is user request for a file download.	60
3. Claim 7.....	61
The computer-implemented method according to claim 5, wherein the detected condition is the detection of a new file attempting to execute.....	61
4. Claim 9.....	62
The computer-implemented method according to claim 1, wherein a determination of whether the executable file is harmful is used to retrain the classifier.	62
5. Claim 26.....	63
The computer-implemented method according to claim 18, wherein the executable file is identified in response to a detected condition.	63
6. Claim 27.....	64

The computer-implemented method according to claim 26, wherein the detected condition is user request for a file download.	64
7. Claim 28.....	64
The computer-implemented method according to claim 26, wherein the detected condition is the detection of a new file attempting to execute....	64
8. Claim 30.....	65
The computer-implemented method according to claim 18, wherein a determination of whether the executable file is harmful is used to retrain the classifier.	65
9. Claim 31.....	65
The computer-implemented method according to claim 18, further comprising preventing execution of the executable file based on classifying the executable file as harmful.....	65
10. Claim 40.....	66
The computer-readable storage device according to claim 32, wherein the executable file is identified in response to a detected condition.	66
11. Claim 41.....	66
The computer-readable storage device according to claim 40, wherein the detected condition is user request for a file download.	66
12. Claim 42.....	67
The computer-readable storage device according to claim 40, wherein the detected condition is the detection of a new file attempting to execute....	67
13. Claim 44.....	67
The computer-readable storage device according to claim 32, wherein a determination of whether the executable file is harmful is used to retrain the classifier.	67

14. Claim 45.....	68
The computer-readable storage device according to claim 32, wherein the process further comprises preventing execution of the executable file based on classifying the executable file as harmful.	68
15. Claim 54.....	68
The system according to claim 46, wherein the executable file is identified in response to a detected condition.	68
16. Claim 55.....	68
The system according to claim 54, wherein the detected condition is user request for a file download.	68
17. Claim 56.....	69
The system according to claim 54, wherein the detected condition is the detection of a new file attempting to execute.	69
18. Claim 58.....	69
The system according to claim 46, wherein a determination of whether the executable file is harmful is used to retrain the classifier.	69
19. Claim 59.....	70
The system according to claim 46, further comprising at least one processor configured to perform operation of: preventing execution of the executable file based on classifying the executable file as harmful.	70
VII. NO BASIS FOR DISCRETIONARY DENIAL.....	71
VIII. CONCLUSION	73

TABLE OF AUTHORITIES

Cases

<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH,</i> IPR2019-01469, Paper 6 (Feb. 13, 2020).....	72
<i>Cellco Partnership v. Huawei Device Co.,</i> IPR2020-01117, Paper 10 (Feb. 3, 2021).....	73
<i>General Electric v. Vestas,</i> IPR2018-00928, Paper 9, *16-17 (PTAB Nov. 5, 2018).....	10
<i>Graham v. John Deere Co.,</i> 383 U.S. 1 (1966)	10
<i>KSR Int’l Co. v. Teleflex, Inc.,</i> 550 U.S. 398 (2007)	10
<i>Phillips v. AWH Corp.,</i> 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	9
<i>Sotera Wireless, Inc., v. Masimo Corp., No.</i> IPR2020-01019, Paper 12 at 19 (P.T.A.B. Dec. 1, 2020).....	71
<i>Western Digital Corporation v. Spex Technologies, Inc.,</i> IPR2018-00084, Paper 14, *11 (PTAB Apr. 25, 2018).....	10

Statutes

35 U.S.C. § 102(a)(1).....	4, 5
35 U.S.C. § 325(d)	71, 73
35 U.S.C. § 314(a)	71
35 U.S.C. § 103.....	passim

Rules

37 C.F.R. § 42.10(a).....	3
37 C.F.R. § 42.104(a).....	4
37 C.F.R. § 42.8(a)(1)	1
37 C.F.R. § 42.8(b)(2).....	1
37 C.F.R. § 42.8(b)(3).....	3

EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 11,409,869 to Schmidler <i>et al.</i> ('869 patent).
1002	Prosecution History of '869.
1003	Expert Declaration of Dr. Seth Nielson.
1004	Curriculum Vitae of A.L. Seth Nielson, Ph.D.
1005	Gil Tahan, <i>Mal-ID: Automatic Malware Detection Using Common Segment Analysis and Meta-Features</i>, Journal of Machine Learning 13, 949-979, (April 13, 2012).
1006	Eitan Menahem, <i>Improving Malware Detection by Applying Multi-Inducer Ensemble</i>, Computational Statistics & Data Analysis 53, (2009).
1007	U.S. Patent Publication No. 2009/0254992 to Schultz <i>et al.</i>
1008	Stuart Russell, <i>Artificial Intelligence: A Modern Approach; The Intelligent Agent Book</i>, 749-751, (Prentice Hall 2) (2003).
1009	Lars Buitinck, <i>API design for machine learning software: Experiences from the scikit-learn project</i>, (September 2013).
1010	Declaration of Ingrid Hsieh-Yee, Ph.D.
1011	Dr. Solomon's Anti-Virus Toolkit Reference Guide (April 1999).
1012	George Karypis, <i>CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling</i>, IEEE 32 (8), (August 1999)
1013	Yingjie Tian, <i>Recent Advances on Support Vector Machine Research</i>, Technological and Economic Development of Economy 18(1), (April 2012).
1014	Ian H. Witten, <i>Data Mining</i>, Practical Machine Learning Tools and Techniques 2, (July 2005).
1015	Craig Szydlowski, <i>Multithreaded Technology & Multicore Processors</i>, Dr. Dobb's Journal, (May 2005).
1016	Webroot, Inc. and Open Text, Inc. v. Trend Micro Inc., Case No. 6:22-cv-00243-ADA-DTG, Claim Construction Order, Dkt. 391 (W.D. Tex., July 25, 2023).
1017	U.S. Patent No. 7,657,838 to Daniell <i>et al.</i>
1018	U.S. Patent No. 8,885,928 to Forman.
1019	U.S. Patent Publication 2010/0082642 to Forman.

Trend Micro, Inc. (“Petitioner”) petitions for *inter partes* review of claims 1-59 of U.S. Patent No. 11,409,869 to Schmidtler *et al.*, entitled “Automatic threat detection of executable files based on static data analysis” (“the ’869 patent,” EX1001), which according to PTO records, is assigned to Webroot Inc.

I. Mandatory Notices (37 C.F.R. § 42.8(a)(1))

REAL PARTY IN INTEREST:

The real parties-in-interest are Trend Micro, Inc., Trend Micro America, Inc. and Trend Micro, Incorporated. Trend Micro, Inc. is a wholly owned subsidiary of Trend Micro America, Inc., which is a wholly owned subsidiary of Trend Micro Incorporated, a Japanese Corporation.

RELATED MATTERS:

Pursuant to 37 C.F.R. § 42.8(b)(2), Petitioner identifies the following related matters.

The ’869 patent is at issue in the following district court litigations:

- *Webroot, Inc. and Open Text, Inc. v. Trend Micro Inc.*, Case No. 6:22-cv-00239 (W.D. Tex.), which was originally filed on March 4, 2022, with an amended complaint listing the ’869 patent filed on September 7, 2022.

- *Webroot, Inc. and Open Text, Inc. v. Sophos LTD.*, Case No. 6:22-cv-00240 (W.D. Tex.), which was originally filed on March 4, 2022, with an amended complaint listing the '869 patent filed on September 7, 2022.
- *Webroot, Inc. and Open Text, Inc. v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.*, Case No. 6:22-cv-00241 (W.D. Tex.), which was originally filed on March 4, 2022, with an amended complaint listing the '869 patent filed on September 7, 2022.
- *Webroot, Inc. and Open Text, Inc. v. AO Kaspersky Lab*, Case No. 6:22-cv-00243 (W.D. Tex.), which was originally filed on March 4, 2022, with an amended complaint listing the '869 patent filed on September 7, 2022.

The '869 patent is a continuation of U.S. Patent No. 10,599,844, which is the subject of the following IPR Petitions:

- *Trend Micro, Inc. v. Webroot, Inc.*, Case No. IPR2023-00662, filed on March 3, 2023.
- *Sophos Ltd. and Sophos Inc., v. Webroot, Inc.*, Case No. IPR2023-00528, filed on February 7, 2023.

LEAD AND BACKUP COUNSEL:

Pursuant to 37 C.F.R. § 42.8(b)(3) and 37 C.F.R. § 42.10(a), Petitioner appoints:

Robert M. Hansen (Reg. No. 43,656) as lead counsel, and

Scott B. Amankwatia (Reg. No. 78,226) as back-up counsel, and

Matthew J. Anderson (Reg. No. 78,257) as back-up counsel.

Pursuant to 37 C.F.R. § 42.10(b), a power of attorney is being filed with this designation of counsel.

SERVICE INFORMATION:

Petitioner provides the following service information for designated counsel:

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, Virginia 20191
Tel: (703)-391-2900
Fax: (703)-391-2900

Petitioner consents to electronic service by email to the email addresses:

rhansen@marburylaw.com;

samankwatia@marburylaw.com

manderson@marburylaw.com; and

pat-docketing@marburylaw.com.

II. Payment of Fees

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 18-1579.

III. Grounds for Standing (37 C.F.R. § 42.104(a))

Petitioner certifies that the '869 patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

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

A. Citation of Prior Art

Petitioner cites the following prior art references:

Mal-ID: Automatic Malware Detection Using Common Segment Analysis and Meta-Features. Gil Tahan, Lior Rokach, Yuval Shahar; 13(Apr):949–979, 2012. (“Tahan” EX1005) is prior art under 35 U.S.C. § 102(a)(1) because it was published and made publicly accessible at least by June 13, 2013, more than one year before the earliest priority date of the '869 patent. EX1010. Tahan was not disclosed to or considered by the USPTO during prosecution of the '869 patent.

Improving malware detection by applying multi-inducer ensemble. Eitan Menahem, Asaf Shabtai, Lior Rokach, Yuval Elovici; Computational Statistics and Data Analysis 53 (2009) 1483–1494. (“Menahem” EX1006) is

prior art under 35 U.S.C. § 102(a)(1)) because it was published and made publicly accessible at least by March 3, 2009, more than one year before the earliest priority date of the '869 patent. EX1001, Cover. Menahem was not disclosed to or considered by the USPTO during prosecution of the '869 patent.

U.S. Patent No. 8,885,928 to Forman ("Forman" EX1018) is prior art under 35 U.S.C. § 102(a)(2) because it was filed on October 25, 2006, which is before the earliest priority date of the '869 patent. EX1018, cover. Forman was not disclosed to or considered by the USPTO during prosecution of the '869 patent. As explained herein, this Forman patent differs from U.S. Patent Publication No. 2010/0082642 to Forman et. al. that was asserted by the Examiner during prosecution.

US Pub 2009/0254992 to Schultz, et al. (Schultz) is prior art under 35 U.S.C. § 102(a)(1) because it published on October 8, 2009, more than one year before the earliest priority date of the '869 patent. EX1005, cover. Schultz was not disclosed to or considered by the USPTO during prosecution of the '869 patent.

B. Statutory Grounds for Challenge

Petitioner requests review of claims 1-59 on the following grounds:

GROUND 1: Claims 1-4, 8, 10-25, 29, 32-39, 43, 46-53, and 57 are unpatentable under 35 U.S.C. § 103 over Tahan (EX1005), Menahem (EX1006) and Forman (EX1018).

GROUND 2: Claims 5-7, 9, 26-28, 30-31, 40-42, 44-45, 54-56, and 58-59 are unpatentable under 35 U.S.C. § 103 over Tahan (EX1005), Menahem (EX1006), Forman (EX1018), and Schultz (EX1007).

V. The '869 Patent

A. Filing Date of the '869 patent

The '869 patent issued 8/9/2022, from Application No. 16/791,649, filed 2/14/2020, which claimed priority as a continuation to Application No. 14/709,875, filed 5/12/2015 (“earliest priority date”), which issued as Patent No. 10,599,844 (“’844 patent”). EX1001, Cover.

B. Specification

The '869 patent describes methods for detecting threats in executable computer files and methods for training probabilistic models that indicate whether executable files are malware or benign by extracting static data points from executable files and applying a trained classifier to the extracted data points. EX1001, Abstract; claim 1. The '869 patent includes claims to computer-implemented methods (claims 1-9), systems implementing the methods of claims

1-4 (claims 10-14) a computer-readable storage device containing instructions implementing the methods of claims 1-3 (claims 15-17), other computer-implemented methods (claims 18-31), a computer-readable storage device containing instructions implementing the methods of claims 18-31 (claims 32-45), and systems implementing the methods of claims 18-31 (claims 46-59).

The specification describes systems and methods related to machine learning classifiers, support vector machines, and ensembles of classifiers. *Id.*, 9:25-50. The systems and methods generate a feature vector from the plurality of static data and use a classifier trained to classify the static data points. *Id.*, 4:29-50. The classifier is trained using files known to be benign or malware and by selectively turning on or off features. *Id.*, 12:61-67; EX1003, ¶96-99.

C. Summary of the Prosecution History

In a first Office Action of 3/29/2021, claims 2-5, 9, 11-16, 18-20¹ were rejected under 35 USC 102(a)(1) based on Shabtai, et al. (“Detection of malicious code by applying machine learning classifiers on static features...”); claims 6-8 were rejected 35 U.S.C. 103 under Shabtai and Basarrate (US 2010/0293273); and claims 10, 17, and 21 were indicated as allowable. EX1002, pp.231-236. After an examiner interview, Patent Owner amended the independent claims to recite

¹ Claim 1 was canceled in a preliminary amendment.

subject matter broader than recited in allowable claim 10. *Id.* pp. 104-138. A

Notice of Allowance (NOA) issued 8/18/2021. *Id.* pp. 178-186.

Patent Owner filed an RCE 11/18/2021 making further amendments, adding claims 22-51, and removing “by a classifier” from the independent claims. *Id.*, pp.163-170. A non-final office action rejected claims 2-5, 9, 11-16, 18-20, 22-28, 33,35, 37-43, 48, and 50 under 35 U.S.C. 103 based on Shabtai and US 2010/0082642 to Forman, et al. (‘642 publication, EX1019),² while claim 10 and related claims were again indicated as allowable. *Id.*, pp.136-142. Patent Owner then amended independent claims 2, 12, 18, 22, and 37 to recite the allowable subject matter of claim 10 without rebutting the Examiner’s assertion that the ‘642 publication discloses turning features on or off for classification, effectively conceding that the ‘642 publication teaches this claim limitation. *Id.*, pp.120-125. Another NOA issued 4/15/2022. *Id.*, pp.81-82.

Patent Owner filed another RCE 5/10/2022 making further amendments and adding claims 52-65 reciting: “wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a

² The ‘642 publication asserted by the Examiner during prosecution is not the same as Forman asserted in Grounds 1 and 2. Forman discloses details discussed in the Grounds not included in the ‘642 publication. However, both Forman and the ‘642 publication disclose turning ON/OFF features for classification.

value of one of the plurality of static data points is within a predetermined range.”
Id., pp.57-65. Another NOA issued 6/15/2022 after which Patent Owner paid the
issue fee. *Id.* pp.19-28.

D. Level of Ordinary Skill in the Art

As established by the Declaration of Dr. Seth Nielson dated September 1,
2023 (*See* EX1003, ¶112), a person having ordinary skill in the art (PHOSITA) as
of the earliest priority date for the ’869 patent would have had a bachelor’s degree
in computer science or a related field, or an equivalent technical degree or
equivalent work experience, and an additional one to two years of education or
experience in the field of computer security topics, such as malware/exploit
detection and/or machine learning. More education can supplement practical
experience and vice versa. EX1003, ¶112.

E. Claim Construction

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); *Phillips v.*
AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

In parallel litigation, Webroot argues claim terms in asserted patents should
be given their plain and ordinary meanings. Likewise, the claim construction order
in this parallel litigation in the Western District of Texas construed contested terms
in the ’869 patent under their plain and ordinary meaning. EX1016, pp.19. For

purposes of this proceeding only, Petitioner proposes the Board adopt plain and ordinary meaning as proposed by Webroot and adopted by the Western District of Texas court. *See Western Digital Corporation v. Spex Technologies, Inc.*, IPR2018-00084, Paper 14, *11 (PTAB Apr. 25, 2018) (finding Petition based on claim constructions urged by Patent Owner satisfies the claim construction requirements and Petitioner is not required to express its subjective agreement regarding correctness of the proffered claim construction or take ownership of the construction); *see also General Electric v. Vestas*, IPR2018-00928, Paper 9, *16-17 (PTAB Nov. 5, 2018).³

VI. Grounds of Rejection

The following discussion and the accompanying evidence demonstrate that there is a reasonable likelihood that Petitioner will prevail with respect to the challenged claims.

Petitioner evaluates the scope and content of the prior art, any differences between the art and the claim, and the knowledge of person of ordinary skill in the art in accordance with *Graham v. John Deere Co.*, 383 U.S. 1 (1966) and *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398 (2007). Petitioner is unaware of any secondary

³ Petitioner reserves the right in other proceedings to construe the Challenged Claims and pursue indefiniteness theories.

considerations relevant to the '869 patent. 550 U.S. at 418.

A. Ground 1: Claims 1-4, 8, 10-25, 29, 32-39, 43, 46-53, and 57 are unpatentable under 35 U.S.C. § 103 over Tahan in view of Menahem and Forman

A PHOSITA would have been motivated to combine the disclosures of Tahan, Menahem and Forman since each relates to machine learning (ML) techniques for identifying malware or spam in files. EX1003, ¶¶114,128-129,144-150,157-183; EX1005, pp. 1-2; EX1006, pp.1-2; EX1018, 2:1-60.

Tahan summarizes the background and design considerations that a PHOSITA would have considered when constructing malware-identifying ML at that time (2009). EX1003, ¶¶159-161; EX1005, pp.1-6. Based on these considerations, Tahan teaches generally-applicable, feature-identifying methods for recognizing malware, including one that arranges relevant executable file segments (e.g., n-grams) into feature vectors that are input to a trained classifier. EX1003, ¶¶117,159-160; EX1005, pp.9-10. Tahan's ML classifiers and techniques are built upon a few detailed examples of systems and techniques known and regularly applied at that time. EX1003, ¶¶116-119,160-161; EX1005, pp.6,9,11. A PHOSITA would have understood the methods in Tahan are combinable and interchangeable based on at least the disclosure in Tahan. EX1003, ¶¶127-128.

Menahem discloses combining different ML classifiers to improve malware identification accuracy, including combining separately-trained classifiers (e.g., trained on file headers, n-grams). EX1006, pp.3-4,7-8; EX1003, ¶¶162-164.

Menahem describes several methods for combining or ensembling specialist sub-classifiers including having inducer classifiers generate meta-features for input into higher-level classifiers as part of a stacked classifier. EX1003, ¶¶162-164; EX1006, pp.4-5,8, Fig.1/Table 2. These combination methods enable separately-trained, specialist classifiers (e.g., Tahan's Mal-ID) to combine their individual capabilities and strengths to obtain more accurate classification of input feature vectors. EX1003, ¶¶162-165; EX1006, pp.1. Accordingly, a PHOSITA would have been motivated to modify the meta-feature analysis of Tahan to generate meta-features with inducers and analyze/select features for input to a classifier to generate a result. EX1003, ¶¶162-165,167-168; EX1006, pp.5-6.

A PHOSITA would have been motivated to combine the predictable, interchangeable base classifiers of Tahan and Menahem to improve the detection of malware as described in Menahem. EX1003, ¶165-170; EX1006, pp.8. A PHOSITA would have recognized the lack of any single "best" classifier as stated in Tahan as a problem and would have been motivated to apply the ensemble methods of Menahem to combine the strengths of the various classifiers in Tahan and Menahem to achieve the desirable results described in Menahem (e.g., by

ensemble of classifiers (also referred to as a super classifier)). EX1003, ¶¶164-170. EX1006, pp.2,4,11. A PHOSITA would have understood that training the base classifiers and ensemble classifiers on different features or different feature formats improves the classification process and provides a more reliable result. EX1003, ¶¶165-166,168-170; EX1006, pp.6.

Further, a PHOSITA would have had a reasonable expectation of success in combining Tahan and Menahem because of significant overlap in the methods of operation and components. EX1003, ¶171. Both references are directed to malware detection by using ML classifiers, and the implementation of Menahem's ensembles and techniques would have been predictable and suggested to PHOSITA at least because Tahan explicitly cites Menahem as a known method for developing classifiers. EX1005, pp.3,30; EX1003, ¶¶167-172.

Likewise, a PHOSITA would have been motivated to modify the stacked classifier including one or more Mal-ID classifiers as disclosed by the Tahan-Menahem combination at least to improve processing and memory efficiency in the classification process. EX1003, ¶¶175,179-180; EX1006, pp.6-7. Forman teaches improving the efficiency of the training and classification processes for machine learning models by eliminating and scaling features to more

efficiently use a larger set of features for classification.⁴ EX1003, ¶175; EX1018, 2:59-3:5, 3:65-4:5, 7:60-67. Further, this scaling disclosed in Forman is compatible and similar in principle to the entropy feature selection of Tahan. EX1003, ¶¶175-178; EX1018, 8:1-21.

A PHOSITA would have found it obvious to include support vector machines (SVMs) in the stacked classifier of Tahan-Menahem in view of the common disclosure of SVMs in Tahan, Menahem and Forman. EX1003, ¶¶174, 184; EX1006, p.3; EX1005, p.3; EX1018, 7:30-55. Indeed, A PHOSITA would have understood that training multiple SVMs and other classifiers (e.g., MAL-ID) on different data and stacking those classifiers would enable the scaling and selection of features as disclosed in Forman to allow the classifier to adapt dynamically at classification. EX1003, ¶¶178-181; EX1018, 7:30-45. Accordingly, the compatible disclosures of Forman would solve a known problem disclosed in Tahan and Menahem. EX1005, pp.9-10; EX1006, p.7; EX1003, ¶¶178-181.

This motivation to combine analysis is incorporated in each subsection / limitation below.

⁴ This teaching of scaling features in Forman is not disclosed in the '642 publication.

1. Claim 1

a. A computer-implemented method comprising:

Tahan discloses a computer-executed algorithm for analyzing a set of files to identify malware. EX1003, ¶114; EX1002, ¶¶187-188. A PHOSITA would have understood that the processes of Tahan were computer-implemented methods. *Id.* Note that claim 18 also recites this pre-amble.

b. identifying, by a knowledge module, static data points that may be indicative of either a harmful or benign executable file;

The '869 patent describes static data points as “a point of reference used to evaluate an executable file” and provides several examples including values and strings extracted from the executable file. EX1001, 5:60-6:1, Table 1.1; EX1003, ¶¶192-195. Likewise, Tahan discloses computationally identifying file segments and 3-grams from an executable file for analysis and recording their corresponding position within the file by means of a repository builder (corresponding to “identifying, by a knowledge module, static data points”). EX1005, pp.7-10; EX1003, ¶196. In a setup phase, Tahan builds repositories of these segments or features that are taken from files that are benign and malware (corresponding to “identifying...static data points that may be indicative of either a harmful or benign executable file”). EX1005, pp.7-8; EX1003, ¶¶196-197. Thus, Tahan teaches this limitation. EX1003, ¶199.

Petitioner notes that independent claims 10, 15, 18, 32, and 46 recite similar limitations absent only “by a knowledge module,” and thus are also taught by Tahan. For economy, arguments addressing claims 10, 15, 18, 32, and 46 limitations reference this section.

c. associating, by the knowledge module, the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;

Tahan discloses that a repository builder (“knowledge module”) collects segments from benign files and malicious files and associates these segments with a benign library (common function library – CFL) or a malware library (threat function library – TFL). EX1005, pp.7-8; EX1003, ¶¶202-203. As explained above, the segments correspond to data points within executable files. Specifically, regarding associating, Tahan discloses:

Note that in the proposed algorithm, we are calculating the distribution of 3-grams within each file and across files, to make sure that a 3-gram belongs to the examined segment and thus associate the segment to either benign (CFL) or malware (TFL).

EX1005, p.7.

In other words, after segmenting files, the repository builder (“knowledge module”) calculates some meta-data (3-grams) and examines them before associating the segment with a benign library or a threat library (corresponding to

“associating... the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files”). EX1005, pp.7-8; EX1003, ¶¶203-205. Thus, Tahan teaches this limitation. EX1003, ¶205.

Petitioner notes that independent claims 10, 15, 18, 32, and 46 recite similar limitations absent only “by a knowledge module,” and thus are also taught by Tahan. For economy, arguments addressing claim 10, 15, 18, 32, and 46 limitations reference this section.

d. identifying an executable file to be evaluated;

Tahan discloses inputting an unclassified file to the detection algorithm to evaluate the file for malware. EX1005, p.10. Further, Tahan teaches analyzing a new application by selecting files of the application for segment analysis. EX1005, p. 8-9; EX1003, ¶¶208-209. Thus, Tahan teaches this limitation. EX1003, ¶210.

Additionally, Menahem discloses inputting files selected for an experimental setting into the ensembled classifier to evaluate which files are malware and which are benign. EX1006, p.8.

Petitioner notes that independent claims 10, 15, 18, 32, and 46 recite similar limitations, and thus are also taught by Tahan. For economy, arguments addressing claims 10, 15, 18, 32, and 46 limitations reference this section.

e. extracting, by the knowledge module, a plurality of static data points from the identified executable file;

Tahan discloses extracting file segments or n-grams from executable files as “features” (corresponding to “static data points”) and recording the corresponding position within each file (corresponding to “extracting... a plurality of static data points from the identified executable file”). EX1005, pp.10-12; EX1003, ¶¶216-218. Tahan’s extracted static data points include the elements of files described in the ’869 patent, such as numeric values, nominal values, strings/byte sequences, and Boolean values. EX1003, ¶¶213-215. Tahan discloses that n-grams or segments represent strings of computer code, variables, or derived values which can identify malware. EX1005, pp.4,7-10; EX1003, ¶¶217-220. These segments can be characterized by meta-features called 3-grams that are derived from the segments. *Id.* Tahan describes that these features and meta-features are initially processed in the same way that is used to build the CFL and TFL repositories. EX1005, pp.9-10; EX1002, ¶¶218-222. Thus, Tahan extracts the segments for unknown files using the same segmenting algorithm (“knowledge module”) as used to segment and aggregate the TFL and CFL from known file categories. *Id.*

Similarly, Menahem teaches processing input sets of files by extracting different types of features including n-grams, portable executable (PE) headers, and other file features (corresponding to “static data points”). EX1003, ¶¶223-227;

EX1006, pp.1,5-6. Specifically, Menahem discloses extracting file information, function-based features based on segments of binary coded functions, numeric values, and character and binary strings. *Id.*

The n-grams and PE header data extracted by Tahan and Menahem would be understood by a PHOSITA to be a plurality of static data points extracted from the identified executable file. EX1003, ¶¶228-230; EX1005, pp.7-9; EX1006, pp.1,5-6.

Thus, Tahan and Menahem both teach this limitation. EX1003, ¶231.

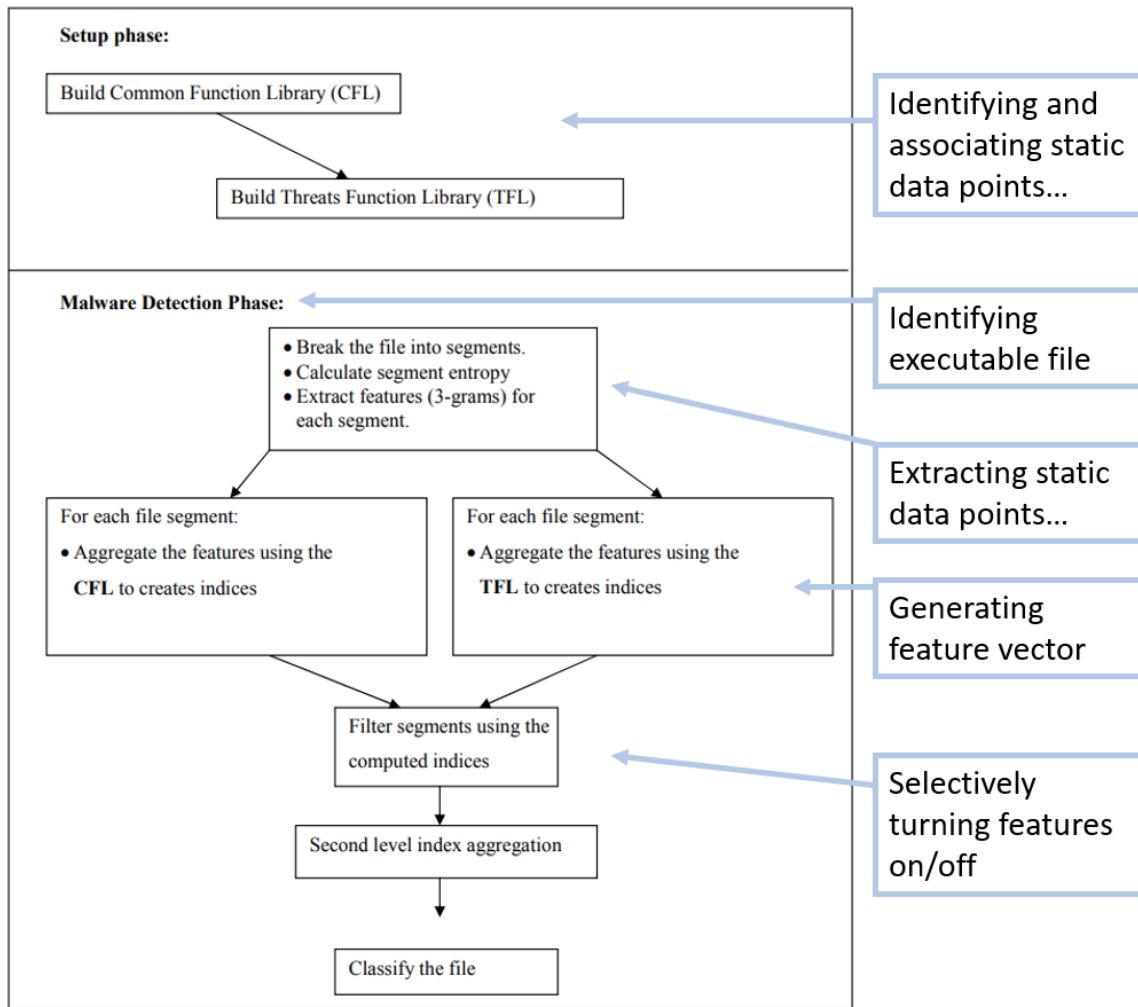
Petitioner notes that independent claims 10, 15, 18, 32, and 46 recite similar limitations absent only “by a knowledge module,” and thus are also taught by Tahan. For economy, arguments addressing claim 10, 15, 18, 32, and 46 limitations reference this section.

f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files,

Tahan discloses generating a classifier, explaining that the machine learning (ML) processes/techniques are performed by a ML mechanism or application. EX1005, pp.9,12-13; EX1003, ¶¶235-237. A POSITA would have understood that ML mechanisms in Tahan disclose classifiers. EX1003, ¶¶235-238 (citing EX1009); EX1005, pp.4,7. This is consistent with the disclosures of the ’869 patent, which sometimes uses the term “classifier” to mean a broader “classifier

component,” as well as a classifier algorithm, which is part of the classifier component. EX1003, ¶¶238-239 (referencing EX1001, 12:48-53, EX1005, EX1009, p.4).

Further, Tahan discloses that this algorithm is trained on the CFL and the TFL, which, as described above regarding limitation 1[b], contain data from known malware and benign files (corresponding to “a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files”). EX1003, ¶¶115-118, 120-123, 242-247, 256-258; EX1005, pp.7,9.



Tahan Figure 1: The Mal-ID method for detecting new malware applications (annotated).

Thus, Tahan's classifier classifies segments ("static data points") in an executable file as malicious or benign or into categories. EX1003, ¶¶245-248,250; EX1005, pp.10-11, Fig.1.

Therefore, a PHOSITA would have understood that Tahan's techniques to generate and optimize the feature vector by Tahan's classifier (which is trained to

classify static data points) discloses “generating a feature vector...using a classifier trained to classify the static data points”. EX1003, ¶¶246-250; EX1005, pp.2-3.

To the extent that the categorized data of Tahan is not input to SVMs, Menahem teaches using known (benign/malware) training data including function-based features as inputs to SVMs for training the classifier. EX1003, ¶¶224-226,259-262; EX1006, pp.6.

Additionally, a PHOSITA would have looked to Menahem for further details to generate feature vectors using machine learning techniques including the meta-features described in Tahan. EX1003, ¶¶251-252,259-260; EX1006, pp.3,13. Further, Forman teaches a method of generating and representing these features for input to a classifier so that processing a large feature set is efficient. EX1018, 2:45-60; EX1003, ¶¶253-254,263. Forman also discloses using and training SVMs based on binary or multiple classes. EX1018, 7:39-45; EX1003, ¶263.

Accordingly, Tahan and Tahan in view of Menahem and Forman teach this limitation. EX1003, ¶¶255,265.

Petitioner notes that independent claims 10, 15, 18, 32, and 46 recite similar limitations, and thus are also taught at least by Tahan. For economy, arguments addressing claims 10, 15, 18, 32, and 46 limitations reference this section.

- g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and.**

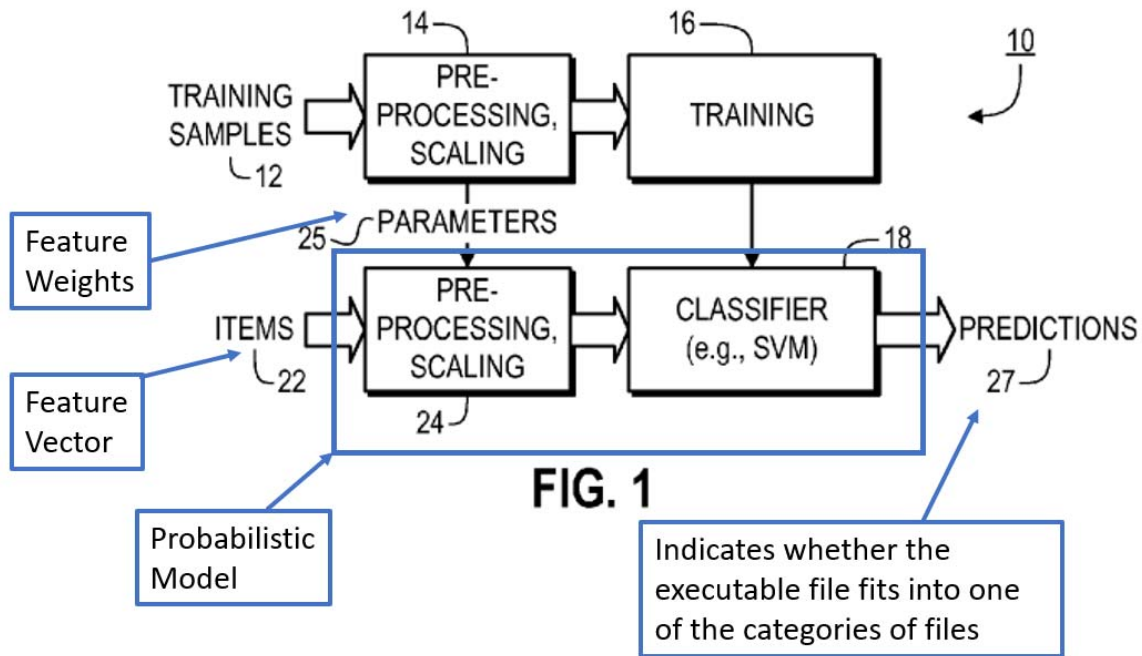
Tahan discloses selecting/deselecting features within feature vectors (corresponding to “selectively turned on or off...”) as part of the process of improving accuracy and efficiency in ML classification of malware. EX1003, ¶267; EX1005, pp.3,8-11. Tahan teaches selecting segments (features / static data points) of files for the feature vector based on an entropy or CFL-MFG index value of the segment, where segments/features are retained or discarded based on whether an entropy value of the segment is within a certain range (e.g., discarded if lower than EntropyLow or higher than EntropyHigh) or whether a CFL-MFG index value is within a certain range (e.g. discarded if CFL-MFG index is larger than zero) (corresponding to “one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range”). EX1003, ¶¶267-271; EX1005, pp.9-10. Tahan discloses that this process of feature selection by entropy is performed as part of an algorithm to “classify a file F as benign or as malware” and is used to eliminate subsets of segments including some which may not have been identified as benign but do not have added value to the

classifier (e.g., a block of white pixels in an image file – which has low entropy).

EX1005, p.10; EX1003, ¶¶267-271.

Further, Tahan teaches that feature selection/reduction during training based on the gain ratio of each segment can increase efficiency and learning speed, Tahan also teaches using this same selection/filtering during classification of files by the trained classifier. EX1005, pp.2-3,8-11, Fig.1 (see annotated above); EX1003, ¶¶267-268, 272. This selection during classification is based on a gain ratio or other similar values, which allow the classifier to judge whether the additional segment adds any signal gain in helping it identify malware or whether the segment is noise. EX1005, p.3,13. A PHOSITA would have understood that a gain ratio or other similar values have ranges, and thus disclose “within a predetermined range”. EX1003, ¶¶272-276. Menahem describes similar compatible selection functionality. EX1003, ¶¶277-282.

Forman teaches integrating scaling and probabilities into the pre-processing of the feature vectors for classification by SVM classifiers.



Forman Figure 1 (annotated)

Forman teaches that feature scaling or elimination as performed during training is also preferably performed during classification.⁵ EX1003, ¶283; EX1018, 7:55-67. Forman also teaches that value ranges can determine elimination of features from a set of features that are input to a classifier for classification. EX1003, ¶¶284-289; EX1018, 8:10-20. Based on these teachings in Forman, a PHOSITA would have understood that the turning on or off of features in a feature vector based on whether one or more static data points is within a predetermined range (e.g., discarding segments as irrelevant to classification as

⁵ This teaching of scaling features in Forman is not disclosed in the '642 publication.

malware based on whether the CFL-MFG index is greater than zero) occurs not just during training, but also during classification of unknown files in order to filter out irrelevant features for efficiency. EX1018, 3:11-14,3:56-60,7:56-60; EX1005, Fig.1,p.10; EX1003, ¶¶283-284.

As noted above, during prosecution Patent Owner did not dispute that the '642 publication discloses turning features on or off during classification. EX1002, pp.138; EX1003, ¶110; see also EX1019, [0045]-[0050]. The portion of the '642 publication cited by the Examiner states:

However, in many cases such an approach would result in too many classifiers being returned. Therefore, in system **500** a subset of the initial features **504** is selected by feature selection module **505**. Such selection preferably is performed based on information concerning index **120**, e.g., information indicating how many classifiers are associated with each feature. Then, for example, in certain embodiments features that are associated with more than a specified threshold number of classifiers (e.g., more than 100, 500, 1,000 or 5,000 classifiers, depending upon the embodiment) are excluded, or the features having the highest number of associated classifiers are excluded until the expected number of classifiers to be returned falls below a specified threshold (although the actual number may differ because multiple different features can be associated with the same classifier). One advantage of the use of a fixed threshold (i.e., the first approach) is that the “stopwords” can be determined in advance for the entire index and then simply filtered out from the extracted features. Of

course, it is possible to simply eliminate those “stopwords” from the index 120 in the first instance; however, doing so would preclude use of the query-widening strategy discussed below.

EX1019, ¶[0047].

In a similar manner, Forman discloses:

As noted above, in certain embodiments the feature set 38 has been modified from an original feature set, e.g., by eliminating certain features. In the preferred embodiments, the training samples 12 have been randomly selected from a group of items to be classified. However, particularly if the items to be classified are relatively static over time and across other groupings, the training samples instead can be selected from a completely separate group than the items that are ultimately to be classified, or from any combination of the same and separate groups.

Next, in step 93 a score is calculated for each of the training samples. In the preferred embodiments, the score comprises the BNS score referenced above. However, as also noted above, in other embodiments a different score is calculated. In certain sub-embodiments, all features are removed that have received a feature score that is zero (or very close to zero based on a pre-defined threshold), because feature values multiplied by zero are always zero and therefore cannot help discriminate positives from negatives.

EX1018, 7:65-8:18.

Accordingly, Tahan in view of Menahem and Forman teaches this limitation. EX1003, ¶292. A PHOSITA would have found this combination

obvious for the reasons noted above in the motivation to combine for this section.

EX1003, ¶¶286-292.

Petitioner notes that independent claims 10, 15, 18, 32, and 46 recite similar limitations, and thus are also taught by Tahan. For economy, arguments addressing claims 10, 15, 18, 32, and 46 limitations reference this section.

h. providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files.

Tahan discloses that support vector machines (SVMs) are a known classification technique that can be trained using reduced feature vectors from benign and malware files to identify malware. EX1005, pp.3; EX1003, ¶¶294-295. In other words, Tahan uses feature vectors input to a classifier that is based on a statistical model to train the classifier (corresponding to “providing the generated feature vector to one or more support vector machines”). EX1005, pp.3-5; EX1003, ¶¶76-80,295; EX1008, p.3 (explaining that SVM classifiers compute a probability or likelihood); EX1013 (describing SVMs with probability calculations).

Likewise, Menahem teaches that its ensemble classifier includes SVMs as base classifiers and builds off prior research that heavily used SVMs as inducers for the meta-feature vectors that are analyzed and then passed on to further classifiers for a final classification. EX1003, ¶¶296-297; EX1006, pp.2-3. A

PHOSITA would have been motivated to incorporate an SVM into the ML classifier ensemble of Menahem to classify executable files together with the classifiers of Tahan at least for performance reasons. EX1003, ¶¶298-300; EX1005, pp.1-3.

Further, Menahem teaches that base-classifiers are individually trained on data from a training set that includes multiple features or attributes. EX1006, pp.7-8; EX1003, ¶296. These individual classifiers include SVMs. EX1006, p.3; EX1003, ¶296. Menahem discloses training the combined classifier, which ensembles the base-classifiers by training various probabilistic weighting factors to select the best base-classifier in different situations (e.g., for different inputs). EX1006, pp.4-7; EX1003, ¶¶297-298. Likewise, when classifying an input as benign/malware, Menahem teaches inputting features from files to be evaluated into SVM base-classifiers and other classifiers. *Id.* The base classifiers then output classifications along with confidence intervals and other assessments by which a final classifier learns and decides which classification to select as the outcome. *Id.* This disclosure corresponds to “providing the generated feature vector to one or more support vector machines to build a probabilistic model”. EX1003, ¶¶297-298. A PHOSITA would have understood how to combine Mal-ID into the composite classifier of Menahem and would have been motivated to do so as explained above. EX1003, ¶¶298-300.

Building on these teachings in Tahan-Menahem, Forman teaches combining a probabilistic weight model with a generated feature vector as inputs to an SVM for classification of a file. EX1003, ¶¶301,307-311; EX1018, 3:24-4:11,4:43-62,7:60-67. That is, the SVM classifier of Forman evaluates a file, and does so by incorporating a probabilistic model related to the feature vector. EX1003, ¶¶309-311. To the extent that Patent Owner argues that Tahan-Menahem do not teach building a probabilistic model, a PHOSITA would have understood that Tahan-Menahem-Forman teaches this limitation. EX1003, ¶¶310-311.

Accordingly, the Tahan-Menahem-Forman combination teaches this limitation. *Id.*

Petitioner notes that independent claims 10 and 15 recite similar limitations, and thus are also taught by Tahan. For economy, arguments addressing claims 10, 15, 18, 32, and 46 limitations reference this section.

For the foregoing reasons, a PHOSITA would have found claim 1 to be obvious based on the Tahan-Menahem-Forman combination. EX1003, ¶312. Accordingly, claim 1 should be found unpatentable over Tahan-Menahem-Forman under 35 U.S.C. § 103.

2. Claim 2

The computer-implemented method according to claim 1, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

Tahan discloses “several novel methods, based on machine learning, to detect malware in executable files without any need for preprocessing, such as unpacking or disassembling.” EX1005, p.1; EX1003, ¶314. A PHOSITA would have understood that Tahan’s processes extract static data points from executable files in a manner that is agnostic to the packing or encryption of the file. EX1003, ¶314. The “MAL-ID” algorithm of Tahan detects malware “with no disassembly, Op-Code analysis, executable header analysis, unpacking nor any other preprocessing...” EX1005, p.27; EX1003, ¶¶314-316. Accordingly, Tahan discloses a method that teaches the claim 2 limitation.

Therefore, a PHOSITA would have found the limitation recited in claim 2 to be obvious based on the disclosures in Tahan. EX1003, ¶317. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 1 from which claim 2 depends. Accordingly, claim 2 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

3. Claim 3

The computer-implemented method according to claim 1, wherein the one or more support vector machines builds the probabilistic model by performing data analysis and pattern recognition on the one or more feature vectors.

As discussed for limitation 1[h], Tahan describes building a probabilistic model including a classification model using data mining techniques. EX1005, pp.4; EX1003, ¶¶294-295,300,319. A PHOSITA would have understood the data mining used in Tahan as being the identification and validation of patterns in data. EX1003, ¶320 (citing EX1014). Thus, a PHOSITA would have understood that such data mining on file features involves performing data analysis and pattern recognition as recited in claim 3. *Id.*

Further, as described regarding limitation 1[h], Forman teaches that a probability distribution can be used to score each feature in a feature set or feature vector that is input into an SVM classifier, where the weights are calculated in training and then supplied in the classification step. EX1018, 4:43-62,7:39-8:21; EX1003, ¶¶321-324.

Thus, a PHOSITA would have found the limitations recited in claim 3 to be obvious based on the Tahan or Tahan in view of Forman. EX1003, ¶¶325-327. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in

claims 1 and 2 from which claim 3 depends. Accordingly, claim 3 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

4. Claim 4

The computer-implemented method according to claim 1, wherein the probabilistic model indicates whether the executable file is harmful.

Tahan and Menahem each disclose or render obvious this limitation. Tahan teaches the use of machine learning techniques to produce a binary classifier that distinguishes malware and benign files. EX1005, pp.2; EX1003, ¶329. The detection algorithm MAL-ID disclosed in Tahan also makes this determination. EX1005, pp.14-16; EX1003, ¶330. A PHOSITA would have understood that the classification of a file as malware by an SVM or MAL-ID as taught in Tahan discloses this claim. EX1003, ¶¶299,306,330.

Menahem discloses that classification results from base classifiers are fed into meta-classifiers in the classifier stack and used to improve probabilities of the overall stack's classification of malware. EX1006, p.5; EX1003, ¶331. As described above, a PHOSITA would have been motivated to combine these disclosures of Tahan and Menahem at least to improve accuracy. EX1003, ¶¶331-332 (citing EX1006, p.4).

Therefore, a PHOSITA would have found the limitation recited claim 4 to be obvious based on the Tahan and Menahem. EX1003, ¶333. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claims 1 and 2

from which claim 4 depends. Accordingly, claim 4 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

5. Claim 8

The computer-implemented method according to claim 1, wherein the plurality of static data points represent predefined character strings in the executable file.

As explained regarding claim 1[e], Tahan teaches extracting segments and/or n-grams from files as features to be input to the classifier. EX1003, ¶335. Further, Tahan teaches that the “segment represents at least one line of code developed using a high-level language” (corresponding to “predefined character strings”) as an example. EX1005, pp.4; EX1003, ¶336. Likewise, the segments are intended to represent meaningful portions of the files since these segments are collected into the CFL and TFL libraries for future reference. EX1005, pp.4,10; EX1003, ¶¶337-338. Menahem also teaches these features. EX1003, ¶¶339-342; EX1006, pp.5-6.

Further, Forman teaches a method for processing text strings and segments into usable and efficient encoding. EX1003, ¶¶343,346. A PHOSITA would have been motivated to apply Forman’s teaching of feature scaling and elimination during classification to tailor over-inclusive feature sets to individual problems and

problem types to achieve goals similar to those of Tahan-Menahem. EX1003, ¶¶344-346.

Therefore, a PHOSITA would have found the limitation recited in claim 8 to be obvious based on the Tahan, Menahem, and Forman. EX1003, ¶347. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 1 from which claim 8 depends. Accordingly, claim 8 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

6. Claim 10

a. A system comprising: at least one memory; and at least one processor operatively connected with the memory and configured to perform operation of:

As claim 10 recites “at least one processor operatively connected with the memory to perform” the claim 1 process, the analysis of claim 1 limitations applies equally to claim 10. EX1003, ¶¶351-352.

Further, Tahan and Menahem disclose that their processes/programs are executed on computers having memory storing computer instructions and algorithms executed by processors. EX1005, pp.14; EX1006, pp.1,6; EX1003, ¶¶352-354.

b. identifying static data points that may be indicative of either a harmful or benign executable file;

As shown re 1[b], Tahan teaches this limitation. EX1003, ¶¶349,355.

- c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;**

As shown re 1[c], Tahan teaches this limitation. EX1003, ¶¶349,355.

- d. identifying an executable file to be evaluated;**

As shown re 1[d], the Tahan-Menahem combination teaches this limitation.

EX1003, ¶¶349,355.

- e. extracting a plurality of static data points from the identified executable file; and**

As shown re 1[e], Tahan and Menahem both teach this limitation. EX1003,

¶¶349,355.

- f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files,**

As shown re 1[f], Tahan and the Tahan-Menahem-Forman combination

teach this limitation. EX1003, ¶¶349,355.

- g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and**

As shown re 1[g], the Tahan-Menahem-Forman combination teaches this limitation. EX1003, ¶¶349,355.⁶

- h. providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files.**

As shown re 1[h], the Tahan-Menahem-Forman combination teaches this limitation. EX1003, ¶¶349,355.

Therefore, a PHOSITA would have found claim 10 to be obvious based on the Tahan-Menahem-Forman combination. Accordingly, claim 10 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

7. Claim 11

The system according to claim 10, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

As shown re claim 2, the Tahan teaches this limitation. EX1003, ¶358. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in

⁶ Note that EX1003 addresses this “wherein” limitation as a third part of limitation 1[f] while this petition addresses this limitation as limitation 1[g].

claim 10 from which claim 11 depends. Accordingly, claim 11 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

8. Claim 12

The system according to claim 10, wherein the one or more support vector machines builds the probabilistic model by performing data analysis and pattern recognition on the one or more feature vectors.

As shown re claim 3, the Tahan teaches this limitation. EX1003, ¶360. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 10 from which claim 12 depends. Accordingly, claim 12 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

9. Claim 13

The system according to claim 10, wherein the probabilistic model indicates whether the executable file is harmful.

As shown re claim 4, the Tahan and Menahem teach this limitation. EX1003, ¶362. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 10 from which claim 13 depends. Accordingly, claim 13 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

10. Claim 14

The system according to claim 10, wherein the plurality of static data points represent predefined character strings in the executable file.

As shown re claim 8, Tahan teaches this limitation. EX1003, ¶364. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 10 from which claim 14 depends. Accordingly, claim 14 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

11. Claim 15

a. A computer-readable storage device containing instructions, that when executed on at least one processor, causing the processor to execute a process comprising:

As claim 15 recites “computer-readable storage device containing instructions, that when executed on at least one processor, causing the processor to execute” the process claim 1, the analysis of claim 1 limitations applies equally to claim 15. EX1003, ¶¶366-367.

Further, Tahan and Menahem disclose that their processes/programs are executed on computers having memory storing computer instructions and algorithms executed by processors. EX1005, pp.14; EX1006, pp.1,6; EX1003, ¶¶368-369.

b. static data points that may be indicative of either a harmful or benign executable file;

As shown re 1[b], Tahan teaches this limitation. EX1003, ¶¶366,370.

- c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;**

As shown re 1[c], Tahan teaches this limitation. EX1003, ¶¶366,370.

- d. identifying an executable file to be evaluated;**

As shown re 1[d], the Tahan-Menahem combination teaches this limitation.

EX1003, ¶¶366,370.

- e. extracting a plurality of static data points from the identified executable file;**

As shown re 1[e], Tahan and Menahem both teach this limitation. EX1003,

¶¶366,370.

- f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files,**

As shown re 1[f], Tahan and the Tahan-Menahem-Forman combination

teaches this limitation. EX1003, ¶¶366, 370.

- g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and**

As shown re 1[g], Tahan and the Tahan-Menahem-Forman combination

teaches this limitation. EX1003, ¶¶366, 370.

- h. providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files.**

As shown re 1[h], the Tahan-Menahem-Forman combination teaches this limitation. EX1003, ¶¶366, 370.

Therefore, a PHOSITA would have found claim 15 to be obvious based on the Tahan-Menahem-Forman combination. Accordingly, claim 15 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

12.Claim 16

The computer-readable storage device according to claim 15, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

As shown re claim 2, Tahan teaches this limitation. EX1003, ¶373. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 15 from which claim 16 depends. Accordingly, claim 16 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

13.Claim 17

The computer-readable storage device according to claim 15, wherein the plurality of static data points represent predefined character strings in the executable file.

As shown re claim 8, Tahan teaches this limitation. EX1003, ¶375. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in

claim 15 from which claim 17 depends. Accordingly, claim 17 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

14.Claim 18

a. A computer-implemented method comprising:

As shown re 1[a], Tahan teaches this preamble. EX1003, ¶377.

b. identifying static data points that may be indicative of either a harmful or benign executable file;

As shown re 1[b], Tahan teaches this limitation. EX1003, ¶¶377,379.

c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;

As shown re 1[c], Tahan teaches this limitation. EX1003, ¶¶377,379.

d. identifying an executable file to be evaluated;

As shown re 1[d], the Tahan-Menahem combination teaches this limitation.

EX1003, ¶¶377,379.

e. extracting a plurality of static data points from the executable file;

As shown re 1[e], Tahan and Menahem both teach this limitation. EX1003,

¶¶377,379.

- f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files,**

As shown re 1[f], Tahan and the Tahan-Menahem-Forman combination both teach this limitation. EX1003, ¶¶377,379.

- g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and**

As shown re 1[g], the Tahan-Menahem-Forman combination teaches this limitation. EX1003, ¶¶377,379.

- h. evaluating the feature vector using a machine learning model to determine whether the executable file fits into one of the categories of files.**

As described regarding 1[h], Tahan, Menahem, and Forman each teach analyzing feature vectors from files of an application using machine learning algorithms (corresponding to “machine learning model”) including support vector machines (SVMs) in order to classify executable files as malware or benign. EX1003, ¶¶381-388; EX1005, p.3; EX1006, p.6.

Therefore, a PHOSITA would have found claim 18 to be obvious based on the Tahan-Menahem-Forman combination. EX1003, ¶389. Accordingly, claim 18 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

15. Claim 19

The computer-implemented method according to claim 18, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

As shown re claim 2, Tahan teaches this limitation. EX1003, ¶391. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 19 depends. Accordingly, claim 19 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

16. Claim 20

The computer-implemented method according to claim 18, wherein the machine learning model comprises an artificial neural network.

Tahan teaches using an artificial neural network (ANN) to classify files as benign/malware. EX1003, ¶393; EX1005, p.3. Tahan teaches that these classification techniques are interchangeable and combinable. EX1005, p. 11. A PHOSITA would have understood how to apply the ANN model of Tahan as described in claim 20. EX1003, ¶394. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 20 depends. Accordingly, claim 20 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

17. Claim 21

The computer-implemented method according to claim 18, wherein the machine learning model comprises a support vector machine.

As shown regarding 1[h], Tahan in view of Menahem teach using an SVM to classify files as benign/malware. EX1003, ¶397; EX1005, pp.3-4; EX1006, p.3. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 21 depends. Accordingly, claim 21 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

18. Claim 22

The computer-implemented method according to claim 18, wherein the machine learning model comprises a machine learning decision tree.

Tahan and Menahem each teach using decision tree learners and decision tree techniques such as C4.5, respectively, as machine learning models to classify files as benign/malware. EX1003, ¶¶400-402; EX1005, p.3; EX1006, p.6. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 22 depends. Accordingly, claim 22 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

19.Claim 23

The computer-implemented method according to claim 18, wherein the machine learning model comprises a Bayesian network.

Tahan, Menahem, and Forman each teach using Naïve-Bayes, Bayesian networks, and other Bayesian methods as interchangeable classification techniques in the context of classifying benign/malware files. EX1003, ¶¶405-407; EX1005, pp.3-4; EX1006, p.6; EX1018, 7:40-45. A POSITA would have understood from these disclosures of Tahan, Menahem, and Forman that many compatible classification techniques exist that are suitable for classifying malware, including Bayesian techniques such as Naïve-Bayes, Multi-Naïve Bayes, and Bayesian networks. EX1003, ¶408. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 23 depends. Accordingly, claim 23 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

20.Claim 24

The computer-implemented method according to claim 18, wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.

Tahan, Menahem, and Forman each teach using support vector machines as a technique for classifying files based on vectors of features in those files. EX1003, ¶¶411-412; EX1005, pp.3-4; EX1006, p.6. Further, the '869 patent describes such

support vector processing as also involving support vector machines. EX1001, 9:25-45,13:7-10; EX1003, ¶411. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 24 depends. Accordingly, claim 24 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

21.Claim 25

The computer-implemented method according to claim 18, wherein the machine learning model comprises a clustering model.

Tahan and Menahem both teach using k-nearest neighbor (KNN) methods as a classification technique in the context of classifying benign/malware files. EX1003, ¶¶415-416; EX1005, pp.3-4; EX1006, p.6. Some clustering models operate in multiple steps and first quantify relatedness using nearest neighbor techniques to discover the clusters. EX1003, ¶417. A POSITA would have known that KNN is a technique used in clustering, a machine learning technique for discovering related groups of data within a data set. EX1003, ¶417 (citing EX1012). Therefore, a PHOSITA would have found claim 25 to be obvious based on the Tahan-Menahem-Forman combination. EX1003, ¶419.

As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 25 depends. Accordingly, claim 25 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

22. Claim 29

The computer-implemented method according to claim 26, wherein the plurality of static data points represent predefined character strings in the executable file.

As shown re claim 8, Tahan teaches this limitation. EX1003, ¶421. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claims 18 and 26 from which claim 29 depends. Accordingly, claim 29 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

23.Claim 32

a. A computer-readable storage device containing instructions, that when executed on a processor, cause the processor to execute a process comprising:

As shown re 1[a] and 15[a], Tahan teaches this preamble. EX1003, ¶¶423,425.

b. identifying static data points that may be indicative of either a harmful or benign executable file;

As shown re 1[b], Tahan teaches this limitation. EX1003, ¶¶423, 426.

c. associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;

As shown re 1[c], Tahan teaches this limitation. EX1003, ¶¶423, 426.

d. identifying an executable file to be evaluated;

As shown re 1[d], the Tahan-Menahem combination teaches this limitation.

EX1003, ¶¶423, 426.

e. extracting a plurality of static data points from the executable file;

As shown re 1[e], Tahan and Menahem both teach this limitation. EX1003,

¶¶423, 426.

f. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files,

As shown re 1[f], Tahan and the Tahan-Menahem-Forman combination both teach this limitation EX1003, ¶¶423, 426.

g. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and

As shown re 1[g], the Tahan-Menahem-Forman combination teaches this limitation. EX1003, ¶¶423, 426.

h. evaluating the feature vector using a machine learning model to determine whether the executable file fits into one of the categories of files.

As shown re 18[h], Tahan, Menahem, and Forman each teach this limitation. EX1003, ¶¶423, 426; EX1005, p.3; EX1006, p.6; EX1018, 7:30-50.

Therefore, a PHOSITA would have found claim 32 to be obvious based on the Tahan-Menahem-Forman combination. EX1003, ¶427. Accordingly, claim 32 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

24.Claim 33

The computer-readable storage device according to claim 32, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

As shown re claims 2 and 19, Tahan teaches this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 33 depends. Accordingly, claim 33 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

25.Claim 34

The computer-readable storage device according to claim 32, wherein the machine learning model comprises an artificial neural network.

As shown re claim 20, Tahan and Menahem both teach this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 34 depends. Accordingly, claim 34 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

26.Claim 35

The computer-readable storage device according to claim 32, wherein the machine learning model comprises a support vector machine.

As shown re claim 21 and claim limitation 1[h], Tahan and Menahem in view of Forman teaches this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 35 depends. Accordingly, claim 35 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

27.Claim 36

The computer-readable storage device according to claim 32, wherein the machine learning model comprises a machine learning decision tree.

As shown re claim 22, Tahan and Menahem both teach this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 36 depends. Accordingly, claim 36 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

28.Claim 37

The computer-readable storage device according to claim 32, wherein the machine learning model comprises a Bayesian network.

As shown re claim 23, Tahan and Menahem both teach this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination

discloses all elements recited in claim 32 from which claim 37 depends.

Accordingly, claim 37 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

29.Claim 38

The computer-readable storage device according to claim 32, wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.

As shown re claim 24 and claim limitation 1[h], the Tahan-Menahem-Forman combination teaches this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 38 depends. Accordingly, claim 38 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

30.Claim 39

The computer-readable storage device according to claim 32, wherein the machine learning model comprises a clustering model.

As shown re claim 25, Tahan and Menahem both teach this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 39 depends. Accordingly, claim 39 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

31.Claim 43

**The computer-readable storage device according to claim 32,
wherein the plurality of static data points represent predefined
character strings in the executable file.**

As shown re claim 29, Tahan teaches this limitation. EX1003, ¶¶429-430.

As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 43 depends. Accordingly, claim 43 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

32.Claim 46

**a. A system comprising: at least one memory; at least one processor
configured to perform operation of:**

As shown re 1[a] and 10[a], Tahan teaches this preamble. EX1003,
¶¶423,425.

**b. identifying static data points that may be indicative of either a
harmful or benign executable file;**

As shown re 1[b], Tahan teaches this limitation. EX1003, ¶¶423,425.

**c. associating the identified static data points with one of a plurality
of categories of files, the plurality of categories of files including
harmful files and benign files;**

As shown re 1[c], Tahan teaches this limitation. EX1003, ¶¶423,425.

d. at least one processor configured to perform operation of:

As shown re 10[a], Tahan teaches this limitation and a PHOSITA would have understood different processes can be assigned to different processors.

EX1003, ¶¶88-92,423,426-427 (citing EX1015).

e. identifying an executable file to be evaluated;

As shown re 1[d], the Tahan-Menahem combination teaches this limitation.

EX1003, ¶¶423,425.

f. extracting a plurality of static data points from the executable file;

As shown re 1[e], Tahan and Menahem both teach this limitation. EX1003,

¶¶423,425.

g. generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files,

As shown re 1[f], Tahan and the Tahan-Menahem-Forman combination both teach this limitation EX1003, ¶¶423,425.

h. wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and

As shown re 1[g], the Tahan-Menahem-Forman combination teaches this limitation. EX1003, ¶¶423,425.

- i. evaluating the feature vector using a machine learning model to determine whether the executable file fits into one of the categories of files.**

As shown re 18[h], Tahan, Menahem, and Forman each teach this limitation.

EX1003, ¶¶423,425; EX1005, p.3; EX1006, p.6.

Therefore, a PHOSITA would have found claim 46 to be obvious based on the Tahan-Menahem-Forman combination. EX1003, ¶427. Accordingly, claim 46 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

33.Claim 47

The system according to claim 46, wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

As shown re claims 2 and 19, Tahan teaches this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 47 depends. Accordingly, claim 47 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

34.Claim 48

The system according to claim 46, wherein the machine learning model comprises an artificial neural network.

As shown re claim 20, Tahan and Menahem both teach this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 48 depends.

Accordingly, claim 48 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

35.Claim 49

The system according to claim 46, wherein the machine learning model comprises a support vector machine.

As shown re 1[h] and claim 21, Tahan in view of Menahem teaches this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 49 depends. Accordingly, claim 49 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

36.Claim 50

The system according to claim 46, wherein the machine learning model comprises a machine learning decision tree.

As shown re claim 22, Tahan and Menahem both teach this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 50 depends. Accordingly, claim 50 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

37.Claim 51

The system according to claim 46, wherein the machine learning model comprises a Bayesian network.

As shown re claim 23, Tahan and Menahem both teach this limitation.

EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 51 depends.

Accordingly, claim 51 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

38.Claim 52

The system according to claim 46, wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.

As shown re 1[h] and claim 24, Tahan, Menahem, and Forman each teach this limitation. EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 52 depends. Accordingly, claim 52 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

39.Claim 53

The system according to claim 46, wherein the machine learning model comprises a clustering model.

As shown re claim 25, Tahan and Menahem both teach this limitation.

EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination

discloses all elements recited in claim 46 from which claim 53 depends.

Accordingly, claim 53 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

40.Claim 57

The system according to claim 46, wherein the plurality of static data points represent predefined character strings in the executable file.

As shown re claim 29, Tahan and Menahem both teach this limitation.

EX1003, ¶¶429-430. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 57 depends.

Accordingly, claim 57 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem and Forman.

B. Ground 2: Claims 5-7, 9, 26-28, 30-31, 40-42, 44-45, 54-56, and 58-59 are unpatentable under 35 U.S.C. § 103 over Tahan in view of Menahem, Forman, and Schultz.

A PHOSITA would have been motivated to combine the disclosures of Tahan-Menahem-Forman with the disclosure of Schultz to increase reliability of identifying malware in files, and to enable active monitoring and protection of a computer system from malware identified by the combined system. EX1003, ¶¶433-435; EX1007, ¶¶[0095]-[0099]. Schultz builds accuracy and breadth for its malware classifiers by utilizing testing and retraining of the classification rules sets. EX1007, ¶¶[0041],[0053],[0085],[0100],[0110],[0112],[0115]; EX1003,

¶¶434-435. Retraining can use newly identified suspicious files from an email trap or from a system administrator to update the classifiers. *Id.* Accordingly, a PHOSITA would recognize that Schultz provides a useful mechanism for recognizing files to be evaluated by the Tahan-Menahem-Forman classification system (e.g., email attachments) and an accuracy-improvement mechanism for updating the Tahan-Menahem-Forman classification system. EX1007, ¶¶[0108]-[0112]; EX1003, ¶434.

A PHOSITA would have been motivated to use the teachings of Schultz to improve on the Tahan-Menahem-Forman methods to identify false positives, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem-Forman. EX1002, ¶435; EX1007, ¶¶[0114]-[0120]. Such a modification would have been predictable to a PHOSITA as of the filing date at least because Tahan and Menahem both cite Schultz as an author producing related work in the field and the methods of Tahan-Menahem-Forman are compatible with those of Schultz. EX1005, pp.5,31; EX1003, ¶¶433; EX1006, pp.3,12.

1. Claim 5

The computer-implemented method according to claim 1, wherein the executable file is identified in response to a detected condition.

Schultz discloses collecting and classifying files that arrive via email, where the analysis is triggered by receiving the file as an attachment in an email from a server. EX1007, ¶[0096]-[0098]; EX1003, ¶437. A PHOSITA would have understood this application of the classifier in Schultz to disclose a condition upon which the executable file is identified to be evaluated as harmful or benign. EX1003, ¶438 (citing EX1001, 7:1-15).

Therefore, a PHOSITA would have found the limitation recited in claim 5 to be obvious based on Schultz. EX1003, ¶¶439-441. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 1 from which claim 5 depends. Accordingly, claim 5 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

2. Claim 6

The computer-implemented method according to claim 5, wherein the detected condition is user request for a file download.

Tahan discloses monitoring files downloaded from the Internet which may be malicious: “Nowadays the use of the Internet has become an integral part of modern life and Internet browsers are downloading to users a wide variety of content, including new computer software. One consequence of this widespread

use is that many computer systems are vulnerable to and infected with malware—malicious software.” EX1005, p.1; EX1003, ¶443. Likewise, as noted regarding claim 5, Schultz teaches filtering and analyzing based on whether an email has an attachment. EX1007, ¶[0104]; EX1003, ¶444. A PHOSITA would have recognized that Tahan and Schultz teach this detected condition and that the processing of Schultz is in response to this detected condition. EX1003, ¶¶445-447.

Therefore, a PHOSITA would have found the limitation recited in claim 6 to be obvious based on Tahan and Schultz. EX1002, ¶448. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claims 1 and 5 from which claim 6 depends. Accordingly, claim 6 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

3. Claim 7

The computer-implemented method according to claim 5, wherein the detected condition is the detection of a new file attempting to execute.

Schultz teaches identifying malicious software in files and programs by performing static analysis and without executing the program. . EX1003, ¶450; EX1007, ¶¶[0017],[0044]; *see supra* claim 5. Further, Schultz teaches quarantining the attachment if it is determined to be malware because Schultz recognizes that without quarantining the attachment may automatically execute itself in the email client. EX1003, ¶451; EX1007, ¶[0014]. Such a quarantine is

necessary because a PHOSITA would recognize that upon release, the attachment could attempt to auto-execute in order to infect the system. EX1003, ¶¶452-454 (citing EX1017, 2:42-50,5:35-39); EX1007, ¶¶[0044],[0103],[0108]. Schultz is also concerned with unknown files and recently received files (via email attachment) (corresponding to “new files”). EX1007, ¶[0123]; EX1003, ¶455. A PHOSITA would have been familiar with the state of the art of anti-malware techniques as of the earliest priority date that included monitoring execution attempts of new files in order to detect and prevent malware. EX1003, ¶456 (citing EX1011).

Therefore, a PHOSITA would have found the limitation recited in claim 7 to be obvious based on Schultz. EX1003, ¶457. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claims 1 and 5 from which claim 7 depends. Accordingly, claim 7 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

4. Claim 9

The computer-implemented method according to claim 1, wherein a determination of whether the executable file is harmful is used to retrain the classifier.

Schultz teaches updating the dataset and classifier rule set (and thus retraining the classifier) based on a determination of whether an executable file is harmful (e.g., a determination made by a classifier that is confirmed as correct).

EX1007, ¶¶[0108]-[0110]; EX1003, ¶¶459-460. In Schultz, the data set is updated with the correct classification, and then the system of Schultz generates a new rule set, which a PHOSITA would have understood to be determining whether a file is harmful or not and retraining the classifier accordingly. *Id.*

Schultz teaches methods of reducing false positives by updating the rule set, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem. EX1003, ¶¶459-462; EX1007, ¶¶[0109]-[0113].

Therefore, a PHOSITA would have found the limitation recited in claim 9 to be obvious based on Schultz. EX1003, ¶463. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 1 from which claim 9 depends. Accordingly, claim 9 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

5. Claim 26

The computer-implemented method according to claim 18, wherein the executable file is identified in response to a detected condition.

As shown re claim 5, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶465. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 26 depends. Accordingly, claim 26 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

6. Claim 27

The computer-implemented method according to claim 26, wherein the detected condition is user request for a file download.

As shown re claim 6, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶467. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claims 18 and 26 from which claim 27 depends. Accordingly, claim 27 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman, and Schultz.

7. Claim 28

The computer-implemented method according to claim 26, wherein the detected condition is the detection of a new file attempting to execute.

As shown re claim 7, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶469. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claims 18 and 26 from which claim 28 depends. Accordingly, claim 28 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

8. Claim 30

The computer-implemented method according to claim 18, wherein a determination of whether the executable file is harmful is used to retrain the classifier.

As shown re claim 9, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶471. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 18 from which claim 30 depends. Accordingly, claim 30 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman, and Schultz.

9. Claim 31

The computer-implemented method according to claim 18, further comprising preventing execution of the executable file based on classifying the executable file as harmful.

Schultz teaches preventing execution of files (e.g., email attachments) including auto-executing attachments when a determined probability value that the executable file is harmful exceeds a threshold. EX1003, ¶¶473-477 (citing EX1017); EX1007, ¶¶[0014],[0095]-[0098],[0103].

Therefore, a PHOSITA would have found the limitation recited in claim 31 to be obvious based on the Tahan-Menahem-Forman-Schultz combination. EX1003, ¶478. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claim 18 from which claim 31 depends.

Accordingly, claim 31 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

10. Claim 40

The computer-readable storage device according to claim 32, wherein the executable file is identified in response to a detected condition.

As shown re claim 26, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claim 32 from which claim 40 depends. Accordingly, claim 40 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman, and Schultz.

11. Claim 41

The computer-readable storage device according to claim 40, wherein the detected condition is user request for a file download.

As shown re claim 27, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claims 32 and 40 from which claim 41 depends. Accordingly, claim 41 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman, and Schultz.

12. Claim 42

**The computer-readable storage device according to claim 40,
wherein the detected condition is the detection of a new file
attempting to execute.**

As shown re claim 28, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claims 32 and 40 from which claim 42 depends. Accordingly, claim 42 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman, and Schultz.

13. Claim 44

**The computer-readable storage device according to claim 32,
wherein a determination of whether the executable file is harmful is
used to retrain the classifier.**

As shown re claim 30, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 44 depends. Accordingly, claim 44 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman, and Schultz.

14. Claim 45

The computer-readable storage device according to claim 32, wherein the process further comprises preventing execution of the executable file based on classifying the executable file as harmful.

As shown re claim 31, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 32 from which claim 45 depends. Accordingly, claim 45 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

15. Claim 54

The system according to claim 46, wherein the executable file is identified in response to a detected condition.

As shown re claim 26, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 54 depends. Accordingly, claim 54 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman, and Schultz.

16. Claim 55

The system according to claim 54, wherein the detected condition is user request for a file download.

As shown re claim 27, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-

Forman-Schultz combination discloses all elements recited in claims 46 and 54 from which claim 55 depends. Accordingly, claim 55 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

17. Claim 56

The system according to claim 54, wherein the detected condition is the detection of a new file attempting to execute.

As shown re claim 28, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-Forman-Schultz combination discloses all elements recited in claims 46 and 54 from which claim 56 depends. Accordingly, claim 56 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

18. Claim 58

The system according to claim 46, wherein a determination of whether the executable file is harmful is used to retrain the classifier.

As shown re claim 30, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-481. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 58 depends. Accordingly, claim 58 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

19. Claim 59

The system according to claim 46, further comprising at least one processor configured to perform operation of: preventing execution of the executable file based on classifying the executable file as harmful.

As shown re claim 31, the Tahan-Menahem-Forman-Schultz combination teaches this limitation. EX1003, ¶¶480-482. Further, Schultz teaches that the filtering, classifying, and quarantining functions described therein are performed by an email processing application that executes on a computer (corresponding to “at least one processor configured to perform operation of: preventing execution”). EX1007, ¶[0020]. To the extent that multiple processors are required by this claim, a PHOSITA would have understood that it was common practice at the time to enumerate or separate operations of a program such as that of Schultz so as to operate on separate processors of a multi-core/processor system. EX1003, ¶¶88-92,482 (citing EX1015, p.2).

Therefore, a PHOSITA would have found the limitation recited in claim 59 to be obvious based on the Tahan-Menahem-Forman-Schultz combination. EX1003, ¶483. As shown, the Tahan-Menahem-Forman combination discloses all elements recited in claim 46 from which claim 59 depends. Accordingly, claim 59 should be found invalid under 35 U.S.C. § 103 based on Tahan, Menahem, Forman and Schultz.

VII. NO BASIS FOR DISCRETIONARY DENIAL

There is no basis to deny institution under 35 U.S.C. §§ 314(a), 325(d).

There is parallel court litigation, so *Fintiv* factors are addressed preliminarily. First, if IPR is instituted, Petitioner stipulates to not raise in the parallel District Court proceedings any ground raised or that could have been reasonably raised in the IPR. *Sotera Wireless, Inc., v. Masimo Corp.*, No. IPR2020-01019, Paper 12 at 19 (P.T.A.B. Dec. 1, 2020) (precedential, designated Dec. 17, 2020). Second, no stay has been entered in the parallel litigation. Third, trial is currently set for August 19, 2024. Trial is currently set for five different cases on that same date in the same Court against Webroot, of which Trend Micro's case is just one. Trial will not proceed on all five cases on the same date, and it is still unclear which, if any, parties will proceed to trial on that date and when trial would be scheduled for the remaining defendants. Thus, it is unclear the exact date Trend Micro's case will go to trial, and it is possible for this inter partes review to reach final written decision before then. Fourth, it is early in the parallel litigation as the parties are still in fact discovery, which is not set to close until January 18, 2024, after which the parties still have expert discovery, dispositive motions, and pretrial disclosures before reaching trial. Finally, Petitioner and Webroot are parties in the parallel litigation as identified in the Mandatory Notices.

Discretionary denial is strongly disfavored here because the “same or substantially the same art or arguments” were not presented to the Office, and the Office “erred in a manner material” to patentability by not asserting the Tahan, Menahem, Forman and Schultz references. *See Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH*, IPR2019-01469, Paper 6, 8-9 (Feb. 13, 2020). Analysis of this issue is guided by the “*Becton*” factors. *Id.* at 9-11.

Tahan, Menahem, Forman and Schultz were not cited, individually or in combination, in any Office communications. *See* EX1006. Regarding *Becton* factors (a) (material differences between asserted art and prior art) and (b) (cumulative nature of the asserted art and prior art), Tahan, Menahem, Forman and Schultz are materially different from the references asserted in Office communications. Thus, the individual references of Tahan, Menahem, Forman and Schultz are not merely cumulative of references the Office relied on. Regarding *Becton* factor (d) (extent of overlap), the arguments in this Petition do not overlap with arguments made during examination. Further, discretionary denial under § 325(d) is still not proper because the Office made a “material error” in not considering the Tahan, Menahem, Forman and Schultz references.

Regarding *Becton* factors (c) (evaluation of art during examination), none of Tahan, Menahem, Forman and Schultz were used in any claim rejections, weighing

strongly against exercising discretion. *See Cellco Partnership v. Huawei Device Co.*, IPR2020-01117, Paper 10, 13 (Feb. 3, 2021).

Regarding *Becton* factors (e) (examiner error) and (f) (other facts), this petition specifically points out and explains how Tahan, Menahem, Forman and Schultz disclose all limitations of the claims.

VIII. CONCLUSION

Inter partes review of claims 1-59 of U.S. Patent No. 11,409,869 is requested based on the grounds described above.

Respectfully submitted,

The Marbury Law Group, PLLC

A handwritten signature in black ink, reading "Robert M. Hansen", written over a horizontal line.

Robert M. Hansen, Reg. No. 43,656
Scott B. Amankwatia, Reg. No. 78,226
Matthew J. Anderson, Reg. No. 78,257

Attorneys for Petitioner

Date: September 5, 2023

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: (703) 391-2900

CERTIFICATION OF SERVICE (37 C.F.R. §§ 42.6(e), 42.105(a))

The undersigned hereby certifies that the above-captioned **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 11,409,869**, Petitioner's power of attorney, this petition for *inter partes* review, and all associated exhibits were served in their entireties on September 5, 2023, on the following party via Federal Express to the Patent Owner by serving to the correspondence address of record for the '869 patent:

**Sprinkle IP Law Group
1301 W. 25th Street, Suite 400
Austin, TX 78705**

A handwritten signature in black ink, reading "Robert M. Hansen", is written over a horizontal line.

Robert M. Hansen, Reg. 43,656
Attorney for Petitioner

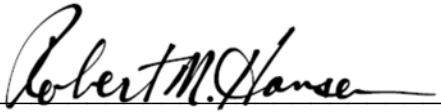
Date: September 5, 2023

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: (703)-391-2900

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION

Pursuant to 37 C.F.R. § 42.24 *et seq.*, the undersigned certifies that this Petition complies with the 14,000-word type-volume limitation. The Petition contains **13,075** words, excluding the table of contents, table of authorities, exhibit list, mandatory disclosures, words in copies of reference figures, signature block, certificates and attached exhibits of the Petition.

The Marbury Law Group, PLLC

A handwritten signature in black ink, reading "Robert M. Hansen", is written over a horizontal line.

Robert M. Hansen, Reg. 43,656
Attorney for Petitioner

Date: September 5, 2023

THE MARBURY LAW GROUP, ^{PLLC}
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191

Tel: 703-391-2900