

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TREND MICRO, INC.
Petitioner

v.

WEBROOT INC.

Case No. IPR2023-00662
Patent No. 10,599,844

**PETITION FOR INTER PARTES REVIEW
OF U.S. PATENT NO. 10,599,844**

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.....	3
III. Grounds for Standing (37 C.F.R. § 42.104(a)).....	3
IV. Identification of Challenge (37 C.F.R. § 42.104(b))	3
A. Citation of Prior Art	3
B. Statutory Grounds for Challenge.....	4
V. The '844 Patent.....	5
A. Filing Date of the '844 patent	5
B. Specification	5
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 1-5, 8, 10-14, 17 and 19 are unpatentable under 35 U.S.C. § 103 over Tahan in view of Menahem	10
1. Claim 1	13
a. A computer-implemented method comprising:	13
b. extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent predefined character strings in the executable file;	13
c. generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable	

files, known benign executable files, and known unwanted executable files, ...	16
d. wherein the collection of data comprises at least a portion of the plurality of static data points, and	19
e. wherein one or more features of the feature vector are selectively turned on or off based on whether a value of one or more static data points from the plurality of extracted static data points is within a predetermined range; and	20
f. evaluating the feature vector using support vector processing to determine whether the executable file is harmful.	22
2. Claim 2	24
The computer-implemented method according to claim 1, wherein the extracting the plurality of static data points further comprises classifying the extracted static data points according to a type of data extracted from the executable file, and encoding the extracted static data points for the classifier based on the classification.	24
3. Claim 3	26
The computer-implemented method according to claim 2, wherein generating the at least one feature vector comprises selectively setting features of the classifier based on the plurality of extracted static data points.....	26
4. Claim 4	28
The computer-implemented method according to claim 2, wherein the classifying the at least one static data point comprises classifying the plurality of extracted static data points into categories comprising	

numeric values, nominal values, string or byte sequences, and Boolean values.....	28
5. Claim 5	29
The computer-implemented method according to claim 4, wherein analyzing the executable file further comprises generating at least one feature vector from the plurality of extracted static data points, wherein features of the generated feature vector are weighted based at least upon the classified categories and the plurality of extracted static data points.	29
6. Claim 8	31
The computer-implemented method according to claim 1, wherein the plurality of static data points is extracted using a machine learning technique.....	31
7. Claim 10	32
a. A computer-readable storage device containing instructions, that when executed on at least one processor, causing the processor to execute a process comprising:	32
b. extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent predefined character strings in the executable file;	33
c..... generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files and known unwanted executable files, ...	33

d.	wherein the collection of data comprises at least a portion of the plurality of static data points, and	34
e.	wherein one or more features of the feature vector are selectively turned on or off based on whether one or more values of one or more static data points from the plurality of extracted static data points is within a predetermined range; and	34
f.	evaluating the feature vector using support vector processing to determine whether the executable file is harmful.	34
8.	Claim 11	35
	The computer-readable storage device according to claim 10, wherein the extracting of the plurality of static data points executed by the processor further comprises classifying the extracted static data points according to a type of data extracted from the executable file, and encoding the extracted static data points for the classifier based on the classifying.	35
9.	Claim 12	37
	The computer-readable storage device according to claim 11, wherein generating the at least one feature vector comprises selectively setting features of the classifier based on the plurality of extracted static data points.....	37
10.	Claim 13	39
	The computer-readable storage device according to claim 11, wherein the classifying the at least one static data point comprises classifying the plurality of extracted static data points into categories comprising numeric values, nominal values, string or byte sequences, and Boolean values.....	39

11. Claim 14	40
The computer-readable storage device according to claim 13, wherein analyzing the executable file further comprises generating a feature vector from the plurality of extracted static data points, wherein features of the generated feature vector are weighted based at least upon the classified categories and the plurality of extracted static data points.	
.....	40
12. Claim 17	42
A system comprising:	
a. at least one memory; and.....	42
b. at least one processor operatively connected with the memory and configured to perform operation of:	42
c. extracting a plurality of predefined character strings from an executable file without decrypting or unpacking the executable file;	42
d. generating a feature vector from the plurality of predefined character strings using a classifier trained to classify the plurality of predefined character strings based on a collection of data comprising known malicious executable files, known benign executable files and known unwanted executable files,.....	46
e. wherein the collection of data comprises at least a portion of one or more of the plurality of predefined character strings, and	49
f. wherein one or more features of the feature vector are selectively turned on or off based on whether a value of one or more predefined character strings from the plurality of predefined character strings is within a predetermined range; and, ...	49

g. evaluating the feature vector using support vector processing to determine whether the executable file is harmful.	51
13. Claim 19	53
The system according to claim 17, wherein the extracting the plurality of predefined character strings further comprises classifying the extracted plurality of predefined character strings according to a type of data extracted from the executable file, and encoding the extracted plurality of predefined character strings for the classifier based on the classification.	53
B. Ground 2: Claims 6, 7, 9, 15-16, 18 and 20 are unpatentable under 35 U.S.C. § 103 as obvious based on Tahan in view of Menahem in view of Schultz.....	55
1. Claim 6	56
The computer-implemented method according to claim 1, wherein determining whether the executable file is harmful further comprises preventing execution of the executable file when a determined probability value that the executable file is harmful exceeds a threshold value.	56
2. Claim 7	58
The computer-implemented method according to claim 6, wherein the threshold value is set based on predetermined false positive range data.	58
3. Claim 9	60
The computer-implemented method according to claim 1, wherein a determination of whether the executable file is harmful is used to retrain the classifier.....	60
4. Claim 15	61

The computer-readable storage device according to claim 10, wherein determining whether the executable file is harmful further comprises preventing execution of the executable file when a determined probability value that the executable file is harmful exceeds a threshold value.....	61
5. Claim 16	63
The computer-readable storage device according to claim 15, wherein the threshold value is set based on predetermined false positive range data.....	63
6. Claim 18	64
The system according to claim 17, wherein the determining further comprises preventing execution of the executable file when a determined probability value that the executable file is harmful exceeds a threshold value, and wherein the threshold value is set based on predetermined false positive range data.....	64
7. Claim 20	67
The system according to claim 17, wherein determining whether the executable file is harmful further comprises preventing execution of the executable file when a determined probability value that the executable file is harmful exceeds a threshold value.....	67
VII. NO BASIS FOR DISCRETIONARY DENIAL.....	69
VIII. CONCLUSION	72

TABLE OF AUTHORITIES

Cases

<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH</i> , IPR2019-01469, Paper 6 (Feb. 13, 2020).....	65
<i>Cellco Partnership v. Huawei Device Co.</i> , IPR2020-01117, Paper 10 (Feb. 3, 2021)	67
<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)	11
<i>KSR Int’l Co. v. Teleflex, Inc.</i> , 550 U.S. 398 (2007)	11, 55, 56
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	9
<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).....	3, 4
35 U.S.C. § 103	passim

Rules

37 C.F.R. § 42.10(a).....	2
37 C.F.R. § 42.104(a).....	3
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).....	2

EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 10,599,844 to Schmidler <i>et al.</i>
1002	Expert Declaration of Dr. Seth Nielson
1003	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).
1004	Eitan Menahem, <i>Improving Malware Detection by Applying Multi-Inducer Ensemble</i>, Computational Statistics & Data Analysis 53, (2009).
1005	U.S. Published Patent Application No. 2009/0254992 to Schultz <i>et al.</i>
1006	<i>Curriculum Vitae</i> of A.L. Seth Nielson, Ph.D.
1007	Prosecution history of U.S. Patent No. 10,599,844
1008	J. Zico Kolter, <i>Learning to Detect and Classify Malicious Executables in the Wild</i>, Journal of Machine Learning Research 7, 2721-2744, (December 2006).
1009	Andrew H. Sung, <i>Intrusion detection using an ensemble of intelligent paradigms</i>, Journal of Network and Computer Applications 28, 167-182, (2005).
1010	Artificial Intelligence, “A Modern Approach, Second Edition” (copyright 2003)
1011	<i>Lars Buitink, Giles Louppe, Mathieu Blondel, et al.</i>, “API design for machine learning software: Experiences from the scikit-learn project”, September 2013.
1012	Y. Ye, L. Chen, D. Wang, T. Li, Q. Jiang, and M. Zhao. Sbmds: an interpretable string based malware detection system using svm ensemble with bagging. Journal in Computer Virology, 5 (4):283–293, 2009.
1013	O. Henchiri and N. Japkowicz. A feature selection and evaluation scheme for computer virus detection. In Data Mining, 2006. ICDM’06. Sixth International Conference on, pages 891–895. IEEE, 2006.

1014	J. Dai, R. Guha, and J. Lee. Efficient virus detection using dynamic instruction sequences. <i>Journal of Computers</i>, 4(5):405–414, 2009.
1015	J. Z. Kolter, “Learning to detect malicious executables in the wild”, In <i>Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining</i>, 2004.
1016	T. Joachims. <i>Advances in kernel methods</i>. chapter Making Large-Scale SVM Learning Practical, pages 169–184. MIT Press, 1999.
1017	Y. Ye, Hierarchical associative classifier (hac) for malware detection from the large and imbalanced gray list. <i>Journal of Intelligent Information Systems</i>, 35 (1):1–20, 2010.
1018	Y. Elovici, A. Shabtai, R. Moskovitch, G. Tahan, and C. Glezer. Applying machine learning techniques for detection of malicious code in network traffic. <i>KI 2007: Advances in Artificial Intelligence</i>, pages 44–50, 2007.
1019	Declaration of Ingrid Hsieh-Yee, Ph.D.

Trend Micro, Inc. (“Petitioner”) petitions for *inter partes* review of claims 1-20 of U.S. Patent No. 10,599,844 to Schmidtler *et al.*, entitled “Automatic threat detection of executable files based on static data analysis” (“the ’844 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 ’844 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.
- *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.

- *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.
- *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.

The '844 patent is also the subject of the following IPR Petition:

- *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 '844 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” EX1003) 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 ’844 patent. EX1019. Tahan was not disclosed to or considered by the USPTO during prosecution of the ’844 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” EX1004) 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 ’844 patent. EX1019. Menahem was not disclosed to or considered by the USPTO during prosecution of the ’844 patent.

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 ’844 patent. EX1005, cover. Schultz was not disclosed to or considered by the USPTO during prosecution of the ’844 patent.

B. Statutory Grounds for Challenge

Petitioner requests review of claims 1-20 on the following grounds:

GROUND 1: Claims 1-5, 8, 10-14, 17 and 19 are unpatentable under 35 U.S.C. § 103 over Tahan (EX1003) and Menahem (EX1004).

GROUND 2: Claims 6-7, 9, 15-16, 18 and 20 are unpatentable under 35 U.S.C. § 103 over Tahan (EX1003), Menahem (EX1004) and Schultz (EX1005).

V. The '844 Patent

A. Filing Date of the '844 patent

The '844 patent issued on 3/24/2020, from U.S. Patent Application No. 14/709,875, filed 5/12/2015 (“filing date”). EX1001, Cover.

B. Specification

The '844 patent describes methods for detecting threats in executable computer files (e.g., software) by extracting static data points from executable files without decrypting or unpacking, and applying a trained classifier to the extracted data points to determine whether the executable file is harmful. EX1001, Abstract. The '844 patent includes claims to computer-implemented methods (claims 1-9), a computer-readable storage device containing instructions implementing the methods (claims 10-16), and systems implementing the methods (claims 17-20).

Claims 1-9 recite computer-implemented methods that include: extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent

predefined character strings in the executable file; generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files, wherein the collection of data comprises at least a portion of the plurality of static data points, and wherein one or more features of the feature vector are selectively turned on or off based on whether a value of one or more static data points from the plurality of extracted static data points is within a predetermined range; and evaluating the feature vector using support vector processing to determine whether the executable file is harmful. EX1001, 16:5-26.

Dependent claims 2-9 recite further limitations for classifying extracted static data points using a classifier (claims 2, 4), preventing execution of executable files when a determined probability value exceeds a threshold value (claims 6 and 7), extracting the static data points using a machine learning technique (claim 8), and using a determination that the executable file is harmful to retrain the classifier (claim 9). *Id.*, 16:27-62.

Claims 10-16 recite a computer-readable storage device on which are stored instructions that when executed on at least one processor cause the processor to execute processes that essentially match the methods recited in claims 1-7. *Id.*, 16:63–18:3. Similarly, claims 17-20 recite a system comprising at least one

memory and at least one processor configured to perform operations that essentially match some of the methods recited in claims 1-6. *Id.*, 18:4-45.

The specification describes systems and methods related to machine learning classifiers, support vector machines, and ensembles of classifiers. *Id.*, 9:40-67.

These classifiers process “feature vectors” of n-grams, portable executable (PE) headers, or numeric values that are extracted from packed and unpacked files and encrypted and unencrypted executable files. *Id.*, 8:49-67. The specification describes methods of training and re-training classifiers and methods of generating a feature vector for input to a trained classifier. *Id.*, 8:10-48; EX1002, ¶¶82,86-87.

C. Summary of the Prosecution History

Application #14/709,875, which issued as the '844 patent, was filed on 5/12/2015 under the first-inventor-to-file provisions of the AIA. EX1007, 394–456. Prosecution began with a non-final Office Action that issued on 12/8/2016, continued through three non-final Office Actions, three final Office Action, five sets of claim amendments, two RCEs and three Examiner interviews, ending in a Notice of Allowance that issued on 11/14/2019. *Id.*, 53–348. The final round of rejections and amendments is illustrative of the Examiner’s findings during the protracted prosecution.

In a Non-final Office Action that issued on 7/17/2019, claims 1-4, 6-13 and 15-20 were rejected, and claims 5 and 14 were indicated to recite allowable subject matter but were objected to for depending from a rejected base claim (i.e., claims 1 and 10, respectively). Specifically, claims 1-2, 10-11 and 19 were rejected as anticipated by non-patent literature prior art and claims 3-4, 6-9, 12-13, 16-18 and 20 were rejected as unpatentable over various combinations of the anticipating reference and five other references. *Id.*, pp.90–99.

In the final amendment filed 9/26/2019, the independent claims were amended to add subject matter recited in claim 5. Specifically, independent claims were amended to recited “... generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files, wherein the collection of data comprises at least a portion of the plurality of static data points, and wherein one or more features of the feature vector are selectively turned on or off based on whether a value of one or more static data points from the plurality of extracted static data points is within a predetermined range....” *Id.*, pp.77–89 (underlines showing the amended language). The Notice of Allowance included an examiner’s statement of reasons for allowance that restated this claim element including the amendment language. *Id.*, pp.18–19.

D. Level of Ordinary Skill in the Art

As established by the Declaration of Dr. Seth Nielson dated March 3, 2023 (See EX1002, ¶94), a person having ordinary skill in the art (PHOSITA) at the time of the earliest claimed priority date for the '844 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. EX1002, ¶94.

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 all claim terms should be given their plain and ordinary meanings. For purposes of this proceeding only, Petitioner proposes the Board adopt Webroot's proposal. 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).¹

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 considerations relevant to the '844 patent. 550 U.S. at 418.

A. Ground 1: Claims 1-5, 8, 10-14, 17 and 19 are unpatentable under 35 U.S.C. § 103 over Tahan in view of Menahem

A PHOSITA would have been motivated to combine the disclosures of Tahan and Menahem since both relate to machine learning (ML) techniques for identifying malware in files. EX1002, ¶¶97,119,130-132. In particular, Tahan begins by providing the background and design considerations that a PHOSITA

¹ Petitioner reserves the right in other proceedings to construe the Challenged Claims and pursue indefiniteness theories.

would have considered when constructing malware-identifying ML at that time. EX1002, ¶¶97-100,130; EX1003, pp.1-6. Tahan's disclosed system itself takes these considerations into account in its design. EX1003, p.7-8; EX1002, ¶¶131-132. Tahan teaches several 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. EX1002, ¶¶96-106,133; EX1003, pp.9-10. Indeed, the disclosed ML classifiers and techniques of Tahan are built upon a few detailed examples of systems and techniques known and regularly applied at that time. EX1002, ¶¶131-132; EX1003, pp.6,9,11. A PHOSITA would have understood that these methods are combinable and interchangeable based on at least the disclosure in Tahan regarding these combinations. EX1002, ¶¶132-135.

Menahem discloses combining different ML classifiers to improve malware identification accuracy, and, in particular, teaches combining separately-trained classifiers (e.g., trained on file headers, n-grams). EX1004, pp.3-4,7-8; EX1002, ¶¶114,120-124. Menahem describes several different methods for combining or ensembling specialist sub-classifiers. EX1002, ¶¶119-121; EX1004, pp.8, Fig.1/Table 2. These combination methods enable the separately-trained, specialist classifiers to combine their individual capabilities and combine their

strengths for a more accurate classification of an input feature vector. EX1002, ¶¶114,121-124,139-141; EX1004, pp.1.

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 and to increase reliability. EX1002, ¶¶138-139; EX1004, 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)). EX1002, ¶¶139-141. EX1004, pp.2,4,11. Likewise, a PHOSITA would have understood that training the base classifiers and the ensemble classifiers on different features or different feature formats improves the classification process and provides a more reliable result. EX1002, ¶¶144-145; EX1004, pp.6. Further, a PHOSITA would have had a reasonable expectation of success in combining Tahan and Menahem because of the significant overlap in the methods of operation and components. EX1002, ¶146. 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. EX1003, pp.3; EX1002, ¶¶143-145. The above motivation to combine analysis is incorporated by reference into each subsection/limitation below.

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, pp.14; EX1002, ¶¶151-153. A PHOSITA as of the filing date would have understood that the processes of Tahan were computer-implemented methods. *Id.*

b. extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent predefined character strings in the executable file;

The '844 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; EX1002, ¶¶157-158. Likewise, Tahan discloses extracting file segments from an executable file as “features” (corresponding to “static data points”) (including those identified by the '844 patent) and recording the corresponding position within the file (corresponding to “extracting a plurality of static data points from an executable file”). EX1003, pp.7-10; EX1002, ¶¶159-171. Tahan’s extracted static data points include types of features identified by the '844 patent such as numeric values,

nominal values, strings/byte sequences, and Boolean values. *Id.* Tahan discloses that n-grams or segments represent strings of computer code, variables, or derived values (corresponding to “wherein the plurality of static data points represent predefined character strings”) which can be used to identify malware. EX1003, pp.4,7-10; EX1002, ¶162-167,193. Tahan discloses that these extraction methods work with encrypted, unencrypted, packed, or unpacked files or other executable files “without any need for preprocessing”, so Tahan does not require decrypting or unpacking an executable file for extraction of features/segments. EX1003, pp.27; EX1002, ¶109,186-189. Specifically, Tahan states:

“[W]e hypothesize that the proposed Mal-ID method performs well in a mixed environment where both obfuscated (including compressed or encrypted) and plain executable files exist... we have independently reconfirmed the validity of the above observation using our method...”
EX1003, pp.27.

A PHOSITA would have understood that the system/method of Tahan can extract static data points from executable files without decrypting or unpacking them. EX1003, ¶188.

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”). EX1002, ¶168-175;

EX1004, 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 (corresponding to “static data points represent predefined character strings”). *Id.* This information may be extracted and packaged by a variety of methods (e.g., n-grams) for analysis by the classifiers disclosed in Menahem. EX1002, ¶¶172-175. Using ensemble methods, Menahem teaches that different features such as n-gram features and PE file header features can be analyzed together within a super or ensemble classifier composed of specialist classifiers. EX1002, ¶¶170,178-183; EX1004, pp.2,12.

During prosecution, in a response filed on April 24, 2019 the Patent Owner argued against the Examiner’s application of U.S. Patent 9,465,940 (Wojnowicz) stating the “non-overlapping file chunks of fixed length” cited by the Examiner were not “predefined character strings”. EX1007, pp.155-156; EX1002, ¶¶193(footnote). To distinguish, the Patent Owner characterized the cited file chunks as “a random (e.g., undefined) character string having no initial significance”. *Id.* As noted above, the segments extracted by the system of Tahan are meaningful portions of those files that correspond to computer code, functions, or values, for example. EX1002, ¶¶186-188; EX1003, pp.2,4,7. In particular, Tahan separates the extracted features into common features shared with benign files and malicious features that can be used as meaningful markers for malware. EX1002,

¶192-194; EX1003, pp.7,10. Moreover, Tahan and Menahem disclose extracting Portable Executable (PE) header fields as static data points, which “represent” predefined character strings because such fields are predefined fields (recognized by the system) with character strings or values represented by character strings. EX1002, ¶¶195-197,167-170,175,177,179,181.

The n-grams and PE header features extracted by Tahan and Menahem would be understood by a PHOSITA as of the filing date to be a plurality of static data points representing predefined character strings that are extractable from an executable file without preprocessing (e.g., decrypting/unpacking) even if it is encrypted/packed. EX1002, ¶¶165-170,183,187-188,194-201; EX1003, pp.7-9; EX1004, pp.1,5-6.

Thus, the combination of Tahan and Menahem teaches this limitation. EX1002, ¶¶156-203,130-149.

- c. generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files,**

Tahan discloses a classifier. Tahan explains that the ML processes/techniques described in its paper are performed by ML mechanism/module/application. EX1003, pp.9,12-13; EX1002, ¶¶100-102. A POSITA would have understood that ML mechanisms / modules / applications in

Tahan disclose classifiers. EX1002, ¶¶112-113,210(citing EX1011),218; EX1003, pp.4,7. This is consistent with the disclosures of the '844 patent, which sometimes uses the term “classifier” to mean a broader “classifier component” (as well as a classifier algorithm, which may be part of the classifier component). EX1002, ¶¶86-88,209-211 (referencing EX1001, 12:48-53, EX1003, EX1011).

Such ML mechanism/module/application (“classifier”) performs the processes discussed in Tahan (including generation of a feature vector from the extracted static data points (as discussed below)). EX1002, ¶¶212-213. Tahan discloses that generating feature vectors and training a classifier to classify using a database of benign, malicious, and unwanted files is fundamental to creation and operation of a classifier (corresponding to “classifier trained...based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files”). EX1002, ¶¶207-208,211; EX1003, pp.3,7. The extracted features of Tahan described regarding claim limitation 1[b] above are disclosed as being compiled together into “a vector containing the features extracted” as part of the first phase of file classification process (corresponding to “generating a feature vector from the plurality of static data points using a classifier trained to...”). EX1002, ¶¶207,212; EX1003, pp.2-3.

Further, Tahan discloses that filtering and building of a feature vector itself may involve ML techniques for filtering and selection, techniques which the '844

patent discloses as being performed by a learning classifier component. EX1002, ¶¶208-217; EX1001, 7:46-56; EX1003, pp.2-3,11. Likewise, Tahan discloses that feature vectors can be composed of meta-features where the features themselves are the result of preliminary classification. EX1002, ¶219; EX1003, pp.9-11,20. Moreover, Tahan's classifier can classify segments ("static data points") in an executable file as malicious or benign, or into categories or data types. EX1002, ¶¶213-219,227; Ex. 1003, pp.4,7,11, Fig.1.

Therefore, a PHOSITA would have understood that these 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 plurality of static data points". *Id.*; EX1002, ¶¶206-228; EX1003, pp.2-3.

Finally, Tahan teaches collecting a common function library from benign files and a threat function library from malware files to have training data comprising known malicious, benign, and unwanted (e.g., spyware) executable files to provide the basis for training the classifier (e.g., to classify data such as files or segments as benign, malware, or spyware) (corresponding to "trained...based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files").

Menahem discloses a similar collection of data. EX1002, ¶¶213-217,229-233; EX1003, pp.2,7-9.

Further to expand on the meta-classifiers and meta-features disclosed in Tahan, Menahem teaches ensembled or stacked classifiers that operate such that the output of one classifier is the input of another classifier. EX1004, pp.4,10,FIG.1; EX1002, ¶¶223-226. As explained above, the higher-level classifiers which receive meta-features are called meta-classifiers by Menahem, such that classifiers in the middle of the hierarchy are trained to classify features extracted from executable files and generate feature vectors for upstream classifiers (e.g., meta-classifiers). EX1004, pp.4,10,FIG.1; EX1002, ¶¶223-226. As stated above, a PHOSITA would have been motivated to combine the classifiers and feature vectors of Tahan with the ensemble techniques of Menahem. EX1002, ¶¶219-226.

Accordingly, Tahan in combination with Menahem teaches this limitation. EX1002, ¶¶130-149,205-235.

d. wherein the collection of data comprises at least a portion of the plurality of static data points, and

Tahan discloses ML training of classifiers on feature vectors that include segments or n-grams of files or functions found in those files (e.g., static data points). EX1002, ¶¶236-238; EX1003, pp.7-9. Such training data sets include benign, malicious, and unwanted files and portions thereof as a collection of data.

Id. Tahan discloses that the training data sets include character strings/segments (static data points) that are expected to appear in at least some benign or malicious files. EX1003, pp.7-9,14-15; EX1002, ¶¶236-238. Accordingly, Tahan discloses the techniques of this limitation as would have been understood by a PHOSITA. EX1002, ¶¶236-239.

- e. wherein one or more features of the feature vector are selectively turned on or off based on whether a value of one or more static data points from the plurality of extracted 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 in ML classification of malware. EX1002, ¶241; EX1003, pp.3. Tahan states:

2. Feature Selection (or feature reduction)—During this phase the vector created in phase 1 is evaluated and redundant and irrelevant features are discarded. Feature selection has many benefits including: improving the performance of learning modules by reducing the number of computations and as a result the learning speed; enhancing generalization capability; improving the interpretability of a model, etc. Feature selection can be done using a wrapper approach or a correlation-based filter approach (Mitchell, 1997). Typically, the filter approach is faster than the wrapper approach and is used when many features exist. The filter approach uses a measure to quantify the correlation of each feature, or a combination of features, to a class. The

overall expected contribution to the classification is calculated and selection is done according to the highest value. The feature selection measure can be calculated using many techniques, such as gain ratio (GR); information-gain (IG); Fisher score ranking technique (Golub et al., 1999) and hierarchical feature selection (Henchiri and Japkowicz, 2006).

EX1003, pp.3.

In other words, Tahan describes various filters and measures (e.g., gain ratio, information gain, Fischer ratio) for selecting or reducing the features in the feature vector being input to the ML classifier. EX1002, ¶¶242-244; EX1003, pp.3,9-11. Additionally, these measures and filters of features (corresponding to “turned on or off based on...a value of one or more static data points”) are checked against thresholds and ranges (corresponding to “predetermined range”) as a part of the selection (e.g., selection by gain ratio quantity). *Id.*

Further, Tahan teaches selecting segments (features / static data points) of a feature vector based on an entropy 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 (corresponding to “one or more features of the feature vector are selectively turned on or off based on whether a value of one or more static data points...is within a predetermined range”). EX1002, ¶¶245-249; EX1003, pp.9-10. Similarly, Menahem teaches using term frequency (and an associated range) in a

file (e.g., TF-IDF) as a measure for selecting features to use to characterize a file as malicious or benign. EX1002, ¶250-255; EX1004, pp.6-7. Accordingly, a PHOSITA would have understood that feature selection/reduction based on values of the static data points (and thus this limitation) is disclosed or rendered obvious by Tahan-Menahem and described as standard practice in Tahan and Menahem. EX1002, ¶130-149,247-259; EX1003, pp.3,10.

f. evaluating the feature vector using support vector processing to determine whether the executable file is harmful.

Tahan discloses using support vector machines (SVM) (which discloses “support vector processing”) to evaluate a feature vector and identify harmful executable files (“determine whether the executable file is harmful”). EX1002, ¶¶264-267; EX1003, pp.2-4. Indeed, Tahan cites a number of academic papers that illustrate that the use of SVM classifiers for malware detection in executable files was commonplace at the earliest possible priority date of the ’844 patent. EX1002, ¶¶112-113,270-271; EX1003, pp.4-5,10. Further, a PHOSITA would have understood the use of SVM as also encompassing the training of the classifier that evaluates the executable file to achieve the claimed “evaluating the feature vector”. EX1002, ¶¶76-79,132; EX1001, 9:40-55.

Menahem teaches that its ensemble classifier includes SVMs as base classifiers and builds off prior researches that heavily used SVMs as inducers for

the meta-feature vectors that are classified and then passed on to further classifiers. EX1002, ¶¶268,273-278; EX1004, 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. EX1002, ¶¶278-279; EX1003, pp.1-3. Thus, a PHOSITA would have found support vector processing for evaluating the feature vector recited in this limitation to be obvious in view of the SVM classifiers disclosed by Tahan or the combination of an SVM into the ensemble ML classifiers of Menahem-Tahan. EX1002, ¶¶264-280(citing EX1015, EX1014, EX1012, EX1013, EX1016); EX1004, pp.4,12; EX1003, pp.3.

Further, both Tahan and Menahem disclose that SVM feature classification is used as part of malware identifying processes and implement SVMs in tests. EX1002, ¶¶269-272; EX1004, pp.2; EX1003, pp.4. Accordingly, a PHOSITA would have understood that the use of SVM in malware determination disclosed in Tahan-Menahem discloses this claim limitation. EX1002, ¶¶130-149,264-281.

For the foregoing reasons, a PHOSITA as of the filing date would have found claim 1 to be obvious based on the Tahan-Menahem combination. EX1002, ¶282. Accordingly, claim 1 should be found unpatentable over the Tahan-Menahem combination under 35 U.S.C. § 103.

2. Claim 2

The computer-implemented method according to claim 1, wherein the extracting the plurality of static data points further comprises classifying the extracted static data points according to a type of data extracted from the executable file, and encoding the extracted static data points for the classifier based on the classification.

Tahan and Menahem both teach generating meta-features of extracted data based on an initial classification with different inducers. EX1002, ¶294; EX1003, pp.9-10,25,28; Ex 1004, p.4-6. Further, Tahan teaches extracting multiple types of features from executable files (e.g., numeric, op-codes, strings, file headers). EX1002, 284-292; EX1003, pp.7-9. As described regarding limitations 1[b]-1[c], the Tahan-Menahem combination discloses classifying static data points and encoding the static data points for a higher-level classifier as meta-features (corresponding to “encoding the extracted static data points for the classifier based on the classification”). EX1003, pp.4,title; EX1004, pp.4; EX1002, ¶292. The essence of a stacked/meta classifier in Menahem involves encoding of classified data as inputs to further classifiers (see specialist classifier), which a PHOSITA would understand to be the encoding limitation of claim 2. EX1002, ¶¶293-299; EX1004, 4,8-11, Fig.1.

Further, Tahan discloses that the extracted n-grams can represent binary features or code segments and can be categorized (corresponding to “classify the extracted static data points according to type of data”) according to value spread

and other aspects of the n-grams. EX1002, ¶¶288-291; EX1003, pp.4-11. Likewise, Menahem extracts PE headers and function-based data including numeric, nominal, and Boolean values and strings as input to base classifiers which then classify and apply classifications/encode as meta-features to upstream classifiers. EX1002, ¶¶292-295; EX1004, pp.4-11. Thus, a PHOSITA would have understood the generation of classified static data points for the classifiers of Tahan and Menahem as classifying and encoding static data points as recited in claim 2. EX1002, ¶¶284-301. Also, a PHOSITA would have understood that Tahan-Menahem's "extracting the plurality of static data points further comprises" the above-discussed encoding and classifying of static data points, or would have been motivated to configure it in such a manner. EX1002, ¶¶130-149,300.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 2 to be obvious based on the Tahan-Menahem combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 1 from which claim 2 depends. Accordingly, claim 2 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

3. Claim 3

The computer-implemented method according to claim 2, wherein generating the at least one feature vector comprises selectively setting features of the classifier based on the plurality of extracted static data points.

As discussed for limitation 1[e], Tahan describes feature selection/reduction as a fundamental practice in malware classification with machine learning.

EX1002, ¶¶303-305; EX1003, pp.3. For example, Tahan implements this feature reduction in the disclosed system as gain ratio feature selection (corresponding to “generating the at least one feature vector comprises selectively setting features”).

EX1002, ¶¶306-308; EX1003, pp.10-13. Further, Tahan notes that features that do not have high entropy are discarded before input into the classifier (corresponding to “selectively setting features of the classifier based on the plurality of extracted

static data points”). EX1002, ¶304; EX1003, pp.10. Finally, Tahan teaches

computing a ranking or score for features extracted from files and building the

feature vector or input to the classifier based on the ranking/score. EX1002, ¶307;

EX1003, pp.3,9. In other words, a PHOSITA would have understood that feature

selection/reduction and feature ranking for input to the classifier discloses the

limitations of claim 3. EX1002, ¶130-149,318.

Menahem discloses similar, compatible ranking and selection of features prior to inputting the features into the classifier, for example, using term frequency in the file. EX1002, ¶309-314; EX1003, pp.5.

Moreover, a PHOSITA would have understood that the above-discussed feature selection/reduction processes of Tahan and Menahem disclose this limitation, including “selectively setting features of the classifier...,” for the reasons discussed above and also because such feature selection/reduction is performed by Tahan-Menahem’s classifier. EX1002, ¶¶315-316.

A PHOSITA would have been motivated to configure the Tahan-Menahem method/system such that “generating the at least one feature vector comprises” the feature selection/reduction processes because they are related processes that involve features and feature vectors. EX1002, ¶¶130-149,315-317.

Thus, a PHOSITA as of the filing date would have found claim 3 to be obvious based on the Tahan-Menahem combination. EX1002, ¶318. As shown, the Tahan-Menahem combination discloses all elements recited in claims 1 and 2 from which claim 3 depends. Accordingly, claim 3 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

4. Claim 4

The computer-implemented method according to claim 2, wherein the classifying the at least one static data point comprises classifying the plurality of extracted static data points into categories comprising numeric values, nominal values, string or byte sequences, and Boolean values.

Tahan and Menahem disclose or render obvious this limitation for the reasons discussed for claim 2. For example, as described regarding claim limitation 1[b] and claim 2, Tahan teaches extracting and classifying strings and header information from files and teaches extracting n-grams for classification, which can include probabilistic ranges (corresponding to “nominal values”), values (corresponding to “numeric values”), and byte sequences (corresponding to “string or byte sequences”). EX1002, ¶¶320-328; EX1003, pp.2-4,26-27. Menahem, likewise, discloses extracting and classifying PE header and function-based static data points, including values, strings, and flags (corresponding to “Boolean values”). EX1002, ¶¶329-332; EX1004, pp.3-6. Menahem extracts features and classifies them in separate feature vectors via base classifiers and, likewise, Tahan develops meta-features of extracted features for input into the classifier. EX1002, ¶332; EX1004, pp.3-4; EX1003, pp.7-12. A PHOSITA would have understood that the example feature types for classification as disclosed in Tahan-Menahem disclose the categories listed in claim 4. EX1002, ¶¶130-149,332-334.

Therefore, a PHOSITA as of the filing date would have found the limitations recited claim 4 to be obvious based on the Tahan-Menahem combination.

EX1002, ¶335. As shown, the Tahan-Menahem combination discloses all elements recited in claims 1 and 2 from which claim 4 depends. Accordingly, claim 4 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

5. Claim 5

The computer-implemented method according to claim 4, wherein analyzing the executable file further comprises generating at least one feature vector from the plurality of extracted static data points, wherein features of the generated feature vector are weighted based at least upon the classified categories and the plurality of extracted static data points.

This “wherein” clause seems to refer to a prior “analyzing” step, but no such step is recited in any of parent claims 1, 2, or 4. Accordingly, this general analysis of the executable file would be interpreted by a PHOSITA as referring to the method of claims 1, 2, and 4 as a whole. EX1002, ¶337. Menahem teaches ensemble classifiers that weight data or features before the features are input into a classifier (corresponding to “features of the generated feature vector are weighted based at least upon...the plurality of extracted static data points”). EX1002, ¶338; EX1004, pp.2-5. Menahem further describes using different types of features (e.g., including numeric, nominal, Boolean, and string) as inputs to inducers so as to

create different behaviors or biases (corresponding to “features of the generated feature vector are weighted based at least upon ... the classified categories”). EX1004, pp.2-3; EX1002, ¶¶338-341. Menahem and Tahan (as discussed for limitation 1[c]) also teach generating feature vectors of the extracted data (corresponding to “generating at least one feature vector from the plurality of extracted static data points”). EX1004, pp.1,5; EX1002, ¶¶342-343. The weighting in the meta-classifiers disclosed in Menahem is based on the parameters and categories of the input classification results. EX1002, ¶¶340-344; EX1004, pp.2-5.

A PHOSITA would have understood that the ensemble classifiers of Tahan-Menahem perform the feature vector generation and weighting of features recited in claim 5. EX1002, ¶343. A PHOSITA would have been motivated to modify Tahan by weighting features of different categories as taught in Menahem to improve performance and the accuracy of the classifiers. EX1004, pp.3-5; EX1002, ¶130-149,344.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 5 to be obvious based on the Tahan-Menahem combination. EX1002, ¶345. As shown, the Tahan-Menahem combination discloses all elements recited in claims 1, 2 and 4 from which claim 5 depends. Accordingly, claim 5 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

6. Claim 8

The computer-implemented method according to claim 1, wherein the plurality of static data points is extracted using a machine learning technique.

Tahan in view of Menahem renders obvious Claim 8. For example, as discussed for limitation 1[e], Tahan and Menahem disclose feature selection/reduction where the process selects/reduces features such that fewer, more relevant features are used in the classification step that determines if an executable file is harmful. EX1002, ¶¶239-260,348; EX1003, pp.3-4,13. There, feature selection/reduction was discussed in the context of selecting/reducing features of a feature vector. *Id.* However, a PHOSITA would have understood that the concepts and techniques of feature selection/reduction (which as discussed below, involves ML techniques) can be applied to and/or implemented in a related context such as for extracting features. EX1002, ¶¶348-351.

For example, a PHOSITA would have understood that the “Gain Ratio” ML technique (for feature selection/reduction) disclosed by Tahan and Menahem can also be applied to the process of extracting features so that fewer, more relevant features can be extracted. EX1002, ¶¶349-350; EX1003, pp.3,9-13; EX1004, pp.4-5. For example, Tahan explains, “The feature selection measure can be calculated using many techniques, such as gain ratio (GR)...” EX1003, pp.3; Ex. 1004, pp.5. A PHOSITA would have been motivated to apply the feature selection/reduction

techniques (including those utilizing ML techniques) to extraction of features, because such a modified extraction process would have resulted in fewer and more relevant features being used in the classification step, and thus would have contributed to the efficiency/effectiveness of the method. EX1002, ¶¶130-149,351-352.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 8 to be obvious based on the Tahan-Menahem combination. EX1002, ¶352. As shown, the Tahan-Menahem combination discloses all elements recited in claim 1 from which claim 8 depends. Accordingly, claim 8 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

7. Claim 10

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

Claim 10 recites a “computer-readable storage device” on which are stored instructions for a processor to perform the process that is recited in claim 1. EX1002, ¶¶354-355. Thus, the analysis of claim 1 limitations applies equally to claim 10 limitations.

Regarding the preamble, Tahan and Menahem disclose that their processes/programs are executed on computers having processors and memory

with computer instructions and algorithms. EX1003, pp.14; EX1002, ¶354-358. A PHOSITA would understand these devices of Tahan and Menahem to encompass computer-readable storage devices as recited in this limitation. EX1002, ¶354-358. A PHOSITA would understand that running a program on a computer with a processor involves having the program instructions be stored on a computer-readable storage medium and executed by the processor to perform the algorithms/processes, such as those of Tahan and Menahem including those discussed for claim 1 and below. *Id.*

b. extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent predefined character strings in the executable file;

As shown above regarding claim 1 limitation b, the combination of Tahan and Menahem teaches this limitation. EX1002, ¶¶156-203,130-149,360-361.

Thus, the combination of Tahan and Menahem teaches this limitation. EX1002, ¶157-201,354.

c. generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files and known unwanted executable files,

As shown above regarding claim 1 limitation c, the combination of Tahan and Menahem teaches this limitation. EX1002, ¶¶130-149,205-235,360-361.

Accordingly, Tahan in combination with Menahem teaches this limitation.

EX1002, ¶¶354,360.

d. wherein the collection of data comprises at least a portion of the plurality of static data points, and

As shown above regarding claim 1 limitation d, the combination of Tahan and Menahem teaches this limitation. EX1002, ¶¶236-239,360-361; EX1003, pp.3,10.

e. wherein one or more features of the feature vector are selectively turned on or off based on whether one or more values of one or more static data points from the plurality of extracted static data points is within a predetermined range; and

As shown above regarding claim 1 limitation e, the combination of Tahan and Menahem teaches this limitation. EX1002, ¶¶130-149,247-259,360-361.

f. evaluating the feature vector using support vector processing to determine whether the executable file is harmful.

As shown above regarding claim 1 limitation f, the combination of Tahan and Menahem teaches this limitation. EX1002, ¶¶130-149,264-281,360-361.

For the forgoing reasons, a PHOSITA as of the filing date would have found claim 10 to be obvious based on the Tahan-Menahem combination. Accordingly, claim 10 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

8. Claim 11

The computer-readable storage device according to claim 10, wherein the extracting of the plurality of static data points executed by the processor further comprises classifying the extracted static data points according to a type of data extracted from the executable file, and encoding the extracted static data points for the classifier based on the classifying.

Tahan and Menahem both teach generating meta-features of extracted data based on an initial classification with different inducers. EX1002, ¶294; EX1003, pp.9-10,25,28; Ex 1004, p.4-6. Further, Tahan teaches extracting multiple types of features from executable files (e.g., numeric, op-codes, strings, file headers). EX1002, 284-292; EX1003, pp.7-9. As described regarding limitations 10[b]-10[d], the Tahan-Menahem combination discloses classifying static data points and encoding the static data points for a higher-level classifier as meta-features (corresponding to “encoding the extracted static data points for the classifier based on the classifying”). EX1003, pp.4,title; EX1004, pp.4; EX1002, ¶157-235,292. The essence of a stacked/meta classifier in Menahem involves encoding of classified data as inputs to further classifiers (see specialist classifier), which a PHOSITA would understand to be the encoding limitation of claim 11. EX1002, ¶¶293-299,362; EX1004, 4,8-11, Fig.1.

Further, Tahan discloses that the extracted n-grams can represent binary features or code segments and can be categorized (corresponding to “classify the

extracted static data points according to type of data”) according to value spread and other aspects of the n-grams. EX1002, ¶¶288-291; EX1003, pp.4-11. Likewise, Menahem extracts PE headers and function-based data including numeric, nominal, and Boolean values and strings as input to base classifiers which then classify and apply classifications/encode as meta-features to upstream classifiers. EX1002, ¶¶292-295,362; EX1004, pp.4-11. Thus, a PHOSITA would have understood the generation of classified static data points for the classifiers of Tahan and Menahem as classifying and encoding static data points as recited in claim 11. EX1002, ¶¶284-301. Also, a PHOSITA would have understood that Tahan-Menahem’s “extracting of...static data points executed by the processor further comprises” the above-discussed encoding and classifying of static data points, or would have been motivated to configure it in such a manner. EX1002, ¶¶130-149,300,362.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 11 to be obvious based on the Tahan-Menahem combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 10 from which claim 11 depends. Accordingly, claim 11 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

9. Claim 12

The computer-readable storage device according to claim 11, wherein generating the at least one feature vector comprises selectively setting features of the classifier based on the plurality of extracted static data points.

As discussed for limitation 10[e], Tahan describes feature selection/reduction as a fundamental practice in malware classification with machine learning. EX1002, ¶¶303-305; EX1003, pp.3. For example, Tahan implements this feature reduction in the disclosed system as gain ratio feature selection (corresponding to “generating the at least one feature vector comprises selectively setting features”). EX1002, ¶¶306-308; EX1003, pp.10-13. Further, Tahan notes that features that do not have high entropy are discarded before input into the classifier (corresponding to “selectively setting features of the classifier based on the plurality of extracted static data points”). EX1002, ¶304; EX1003, pp.10. Finally, Tahan teaches computing a ranking or score for features extracted from files and building the feature vector or input to the classifier based on the ranking/score. EX1002, ¶307; EX1003, pp.3,9. In other words, a PHOSITA would have understood that feature selection/reduction and feature ranking for input to the classifier discloses the limitations of claim 12. EX1002, ¶318,362.

Menahem discloses similar, compatible ranking and selection of features prior to inputting the features into the classifier, for example, using term frequency in the file. EX1002, ¶309-314; EX1003, pp.5.

Moreover, a PHOSITA would have understood that the above-discussed feature selection/reduction processes of Tahan and Menahem disclose this limitation, including “selectively setting features of the classifier...,” for the reasons discussed above and also because such feature selection/reduction is performed by Tahan-Menahem’s classifier. EX1002, ¶¶315-316,362.

A PHOSITA would have been motivated to configure the Tahan-Menahem method/system such that “generating the at least one feature vector comprises” the feature selection/reduction processes because they are related processes that involve features and feature vectors. *Id.* ¶130-149,317,362.

Thus, a PHOSITA as of the filing date would have found the limitations recited in claim 12 to be obvious based on the Tahan-Menahem combination. As shown, the Tahan-Menahem combination discloses all elements recited in claims 10 and 11 from which claim 12 depends. Accordingly, claim 12 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

10. Claim 13

The computer-readable storage device according to claim 11, wherein the classifying the at least one static data point comprises classifying the plurality of extracted static data points into categories comprising numeric values, nominal values, string or byte sequences, and Boolean values.

Tahan and Menahem disclose or render obvious this limitation for the reasons discussed for claim 11. For example, as described regarding claim limitation 10[b] and claim 11, Tahan teaches extracting and classifying strings and header information from files and teaches extracting n-grams for classification, which can include probabilistic ranges (corresponding to “nominal values”), values (corresponding to “numeric values”), and byte sequences (corresponding to “string or byte sequences”). EX1002, ¶¶320-328,362; EX1003, pp.2-4,26-27. Menahem, likewise, discloses extracting and classifying PE header and function-based static data points, including values, strings, and flags (corresponding to “Boolean values”). EX1002, ¶¶329-332; EX1004, pp.3-6. Menahem extracts features and classifies them in separate feature vectors via base classifiers and, likewise, Tahan develops meta-features of extracted features for input into the classifier. EX1002, ¶332,362; EX1004, pp.3-4; EX1003, pp.7-12. A PHOSITA would have understood that the example feature types for classification as disclosed in Tahan-Menahem disclose the categories listed in claim 13. EX1002, ¶¶130-149,332-334,362.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 13 to be obvious based on the Tahan-Menahem combination. As shown, the Tahan-Menahem combination discloses all elements recited in claims 10 and 11 from which claim 13 depends. Accordingly, claim 13 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

11. Claim 14

The computer-readable storage device according to claim 13, wherein analyzing the executable file further comprises generating a feature vector from the plurality of extracted static data points, wherein features of the generated feature vector are weighted based at least upon the classified categories and the plurality of extracted static data points.

This “wherein” clause seems to refer to a prior “analyzing” step, but no such step is recited in any of the parent claims 10, 11, or 13. Accordingly, this general analysis of the executable file would be interpreted by a PHOSITA as referring to the instructions on the computer-readable storage device of claims 10, 11, and 13 as a whole. EX1002, ¶337. Menahem teaches ensemble classifiers that weight data or features before the features are input into a classifier (corresponding to “features of the generated feature vector are weighted based at least upon...the plurality of extracted static data points”). EX1002, ¶338,362; EX1004, pp.2-5. Menahem further describes using different types of features (e.g., including numeric,

nominal, Boolean, and string) as inputs to inducers so as to create different behaviors or biases (corresponding to “features of the generated feature vector are weighted based at least upon ... the classified categories”). EX1004, pp.2-3; EX1002, ¶¶338-341. Menahem and Tahan (as discussed for limitation 10[d]) also teach generating feature vectors of the extracted data (corresponding to “generating at least one feature vector from the plurality of extracted static data points”). EX1004, pp.1,5; EX1002, ¶¶342-343. The weighting in the meta-classifiers disclosed in Menahem is based on the parameters and categories of the input classification results. EX1002, ¶¶340-344,362; EX1004, pp.2-5.

A PHOSITA would have understood that the ensemble classifiers of Tahan-Menahem perform the feature vector generation and weighting of features recited in claim 14. EX1002, ¶343. A PHOSITA would have been motivated to modify Tahan by weighting features of different categories as taught in Menahem to improve performance and the accuracy of the classifiers. EX1004, pp.3-5; EX1002, ¶¶130-149,344,362.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 14 to be obvious based on the Tahan-Menahem combination. As shown, the Tahan-Menahem combination discloses all elements recited in claims 10, 11 and 13 from which claim 14 depends. Accordingly, claim 14 should be

found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

12. Claim 17

A system comprising:

Tahan and Menahem disclose operating the malware detection processes on computer systems. EX1002, ¶¶367-369; EX1003, pp.14. A PHOSITA would understand the systems of Tahan and Menahem to correspond to the system recited in claim 17. *Id.*; EX1002, ¶322-323.

a. at least one memory; and

Tahan and Menahem disclose a computer having memory (e.g., 4GB of RAM) that is a part of the experimental setup. EX1002, ¶371-372; EX1003, pp.14; EX1004, p.5.

b. at least one processor operatively connected with the memory and configured to perform operation of:

Tahan and Menahem disclose a computer having a CPU or processor that is a part of the experimental setup with the memory. EX1002, ¶371-372; EX1003, pp.14; EX1004, p.5.

c. extracting a plurality of predefined character strings from an executable file without decrypting or unpacking the executable file;

Tahan discloses extracting file segments from an executable file as “features” and recording the corresponding position within the file (corresponding

to “extracting a plurality of predefined character strings from an executable file”).² EX1003, pp.7-10; EX1002, ¶159-171. Tahan’s extracted strings or segments include types of features identified by the ’844 patent as being extracted such as numeric values, nominal values, strings/byte sequences, and Boolean values. *Id.*; EX1001, Table1.1. Tahan discloses that n-grams or segments represent strings of computer code, variables, or derived values (corresponding to “predefined character strings”) which can be used to identify malware. EX1003, pp.4,7-10; EX1002, ¶162-167,193,365,374.

Tahan discloses that these extraction methods work with encrypted, unencrypted, packed, or unpacked files or other executable files “without any need for preprocessing”, so Tahan does not require decrypting or unpacking an executable file for extraction of features/segments. EX1003, pp.27; EX1002, ¶109,186-189. Specifically, Tahan states:

[W]e hypothesize that the proposed Mal-ID method performs well in a mixed environment where both obfuscated (including compressed or

² As discussed above for claim 1, Tahan and Menahem’s features (that are extracted from files) disclose static data points that represent predefined character strings. Tahan/Menahem’s disclosures relating to such features also apply to and disclose or render obvious predefined character strings, so the analysis for claim 1 applies to claim 17. EX1002, ¶374.

encrypted) and plain executable files exist... we have independently reconfirmed the validity of the above observation using our method.... EX1003, pp.27.

A PHOSITA would have understood that the system/method of Tahan can extract features/character strings from executable files without decrypting or unpacking them. EX1003, ¶188,365.

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 “predefined character strings”). EX1002, ¶168-175; EX1004, 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.* This information may be extracted and packaged by a variety of methods (e.g., n-grams) for analysis by the classifiers disclosed in Menahem. EX1002, ¶¶172-175,365,374. Using ensemble methods, Menahem teaches that different features such as n-gram features and PE file header features can be analyzed together within a super or ensemble classifier composed of specialist classifiers. EX1002, ¶170,178-183; EX1004, pp.2,12.

During prosecution, in a response filed on April 24, 2019 the Patent Owner argued against the Examiner’s application of U.S. Patent 9,465,940 (Wojnowicz) stating the “non-overlapping file chunks of fixed length” cited by the Examiner

were not “predefined character strings”. EX1007, pp.155-156; EX1002, ¶193(footnote). To distinguish, the Patent Owner characterized the cited file chunks as “a random (e.g., undefined) character string having no initial significance”. *Id.* As noted above, the segments extracted by the system of Tahan are meaningful portions of those files that correspond to computer code, functions, or values, for example. EX1002, ¶186-188; EX1003, pp.2,4,7. In particular, Tahan separates the extracted features into common features shared with benign files and malicious features that can be used as meaningful markers for malware. EX1002, ¶192-194; EX1003, pp.7,10. Moreover, Tahan and Menahem disclose extracting Portable Executable (PE) header fields as features which disclose predefined character strings because such fields are predefined fields (recognized by the system) with character strings or values represented by character strings. EX1002, ¶¶195-197,167-170,175,177,179,181.

The n-grams and PE header features extracted by Tahan and Menahem would be understood by a PHOSITA as of the filing date to be predefined character strings that are extractable from an executable file without preprocessing (e.g., decrypting/unpacking) even if it is encrypted/packed. EX1002, ¶¶165-170,183,187-188,194-201; EX1003, pp.7-9; EX1004, pp.1,5-6.

Thus, the combination of Tahan and Menahem teaches this limitation. EX1002, ¶¶130-149,156-203,130-149,365,374.

- d. generating a feature vector from the plurality of predefined character strings using a classifier trained to classify the plurality of predefined character strings based on a collection of data comprising known malicious executable files, known benign executable files and known unwanted executable files,**

Tahan discloses a classifier. Tahan explains that the ML processes/techniques described in its paper are performed by ML mechanism / module/application. EX1003, pp.9,12-13; EX1002, ¶¶100-102. A POSITA would have understood that ML mechanisms/modules/applications in Tahan disclose classifiers. EX1002, ¶¶112-113,210(citing EX1011),218; EX1003, pp.4,7. This is consistent with the disclosures of the '844 patent, which sometimes uses the term “classifier” to mean a broader “classifier component” (as well as a classifier algorithm, which may be part of the classifier component). EX1002, ¶¶86-88,209-211 (referencing EX1001, 12:48-53, EX1003, EX1011),365,374.

Such ML mechanism/module/application (“classifier”) performs the processes discussed in Tahan (including generating of a feature vector from the predefined character strings). EX1002, ¶¶212-213. Tahan discloses that generating feature vectors and training a classifier to classify using a database of benign, malicious, and unwanted files is fundamental to creation and operation of a classifier (corresponding to “classifier trained...based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files”). EX1002, ¶¶207-208,211; EX1003, pp.3,7. The

extracted features of Tahan described regarding claim limitation 17[c] above are disclosed as being compiled together into “a vector containing the features extracted” as part of the first phase of file classification process (corresponding to “generating a feature vector from the plurality of predefined character strings using a classifier trained to...”). EX1002, ¶¶207,212,365,374; EX1003, pp.2-3.

Further, Tahan discloses that filtering and building of a feature vector itself may involve ML techniques for filtering and selection, techniques which the '844 patent discloses as being performed by a learning classifier component. EX1002, ¶¶208-217; EX1001, 7:46-56; EX1003, pp.2-3,11. Likewise, Tahan discloses that feature vectors can be composed of meta-features where the features themselves are the result of preliminary classification. EX1002, ¶219; EX1003, pp.9-11,20. Moreover, Tahan's classifier can classify segments/features (“predefined character strings”) in an executable file as malicious or benign, or into categories or data types. EX1002, ¶¶213-219,227,374; Ex. 1003, pp.4,7,11, Fig.1.

Therefore, a PHOSITA would have understood that these techniques to generate and optimize the feature vector by Tahan's classifier (which is trained to classify predefined character strings) discloses “generating a feature vector...using a classifier trained to classify the plurality of predefined character strings”. *Id.*; EX1002, ¶¶206-228,365,374; EX1003, pp.2-3.

Finally, Tahan teaches collecting a common function library from benign files and a threat function library from malware files to have training data comprising known malicious, benign, and unwanted (e.g., spyware) executable files to provide the basis for training the classifier (e.g., to classify data such as files or segments as benign, malware, or spyware) (corresponding to “trained...based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files”). Menahem discloses a similar collection of data. EX1002, ¶¶213-217,229-233,365; EX1003, pp.2,7-9.

Further to expand on the meta-classifiers and meta-features disclosed in Tahan, Menahem teaches ensembled or stacked classifiers that operate such that the output of one classifier is the input of another classifier. EX1004, pp.4,10,FIG.1; EX1002, ¶223-226. As explained above, the higher-level classifiers which receive meta-features are called meta-classifiers by Menahem, such that classifiers in the middle of the hierarchy are trained to classify features extracted from executable files and generate feature vectors for upstream classifiers (e.g., meta-classifiers). EX1004, pp.4,10,FIG.1; EX1002, ¶¶223-226. As stated above, a PHOSITA would have been motivated to combine the classifiers and feature vectors of Tahan with the ensemble techniques of Menahem. EX1002, ¶¶219-226,374.

Accordingly, Tahan in combination with Menahem teaches this limitation.

EX1002, ¶¶130-149,205-235,365,374.

e. wherein the collection of data comprises at least a portion of one or more of the plurality of predefined character strings, and

Tahan discloses ML training of classifiers on feature vectors that include segments or n-grams of files or functions found in those files (e.g., predefined character strings). EX1002, ¶¶236-238,365,374; EX1003, pp.7-9. Such training data sets include benign, malicious, and unwanted files and portions thereof as a collection of data. *Id.* Tahan discloses that the training data sets include character strings/segments/features (predefined character strings) that are expected to appear in at least some benign or malicious files. EX1003, pp.7-9,14-15; EX1002, ¶¶236-238,365,374. Accordingly, Tahan discloses the techniques of this limitation as would have been understood by a PHOSITA. EX1002, ¶¶236-239,365,374.

f. wherein one or more features of the feature vector are selectively turned on or off based on whether a value of one or more predefined character strings from the plurality of predefined character strings 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 in ML classification of malware. EX1002, ¶241; EX1003, pp.3.

Tahan states:

2. Feature Selection (or feature reduction)—During this phase the vector created in phase 1 is evaluated and redundant and irrelevant features are discarded. Feature selection has many benefits including: improving the performance of learning modules by reducing the number of computations and as a result the learning speed; enhancing generalization capability; improving the interpretability of a model, etc. Feature selection can be done using a wrapper approach or a correlation-based filter approach (Mitchell, 1997). Typically, the filter approach is faster than the wrapper approach and is used when many features exist. The filter approach uses a measure to quantify the correlation of each feature, or a combination of features, to a class. The overall expected contribution to the classification is calculated and selection is done according to the highest value. The feature selection measure can be calculated using many techniques, such as gain ratio (GR); information-gain (IG); Fisher score ranking technique (Golub et al., 1999) and hierarchical feature selection (Henchiri and Japkowicz, 2006).

EX1003, pp.3.

In other words, Tahan describes various filters and measures (e.g., gain ratio, information gain, Fischer ratio) for selecting or reducing the features in the feature vector being input to the ML classifier. EX1002, ¶¶242-244,365,374; EX1003, pp.3,9-11. Additionally, these measures and filters of features (corresponding to “turned on or off based on...a value of one or more predefined character strings”)

are checked against thresholds and ranges (corresponding to “predetermined range”) as a part of the selection (e.g., selection by gain ratio quantity). *Id.*

Further, Tahan teaches selecting segments (features / predefined character strings) of a feature vector based on an entropy 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 (corresponding to “one or more features of the feature vector are selectively turned on or off based on whether a value of one or more predefined character strings...is within a predetermined range”). EX1002, ¶¶245-249,365,374; EX1003, pp.9-10. Similarly, Menahem teaches using term frequency (and an associated range) in a file (e.g., TF-IDF) as a measure for selecting features to use to characterize a file as malicious or benign. EX1002, ¶¶250-255; EX1004, pp.6-7. Accordingly, a PHOSITA would have understood that feature selection/reduction based on values of the predefined character strings (and thus this limitation) is disclosed or rendered obvious by Tahan-Menahem and described as standard practice in Tahan/Menahem. EX1002, ¶¶130-149,247-259,374; EX1003, pp.3,10.

g. evaluating the feature vector using support vector processing to determine whether the executable file is harmful.

Tahan discloses using support vector machines (SVM) (which discloses “support vector processing”) to evaluate a feature vector and identify harmful

executable files (“determine whether the executable file is harmful”). EX1002, ¶¶264-267; EX1003, pp.2-4. Indeed, Tahan cites a number of academic papers that illustrate that the use of SVM classifiers for malware detection in executable files was commonplace at the earliest possible priority date of the ’844 patent. EX1002, ¶¶112-113,270-271,365,374; EX1003, pp.4-5,10; *see also* EX1008,EX1009, EX1010,EX1011,EX1012,EX1013,EX1014,EX1015,EX1016,EX1017,EX1018. Further, a PHOSITA would have understood the use of SVM as also encompassing the training of the classifier that evaluates the executable file to achieve the claimed “evaluating the feature vector”. EX1002, ¶¶76-79,132,374; EX1001, 9:40-55.

Menahem teaches that its ensemble classifier includes SVMs as base classifiers and builds off prior researches that heavily used SVMs as inducers for the meta-feature vectors that are classified and then passed on to further classifiers. EX1002, ¶¶268,273-278; EX1004, 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. EX1002, ¶¶278-279,374; EX1003, pp.1-3. Thus, a PHOSITA would have found support vector processing for evaluating the feature vector recited in this limitation to be obvious in view of the SVM classifiers disclosed by Tahan or the combination of an SVM into the ensemble ML

classifiers of Menahem-Tahan. EX1002, ¶¶264-280(citing EX1015, EX1014, EX1012, EX1013, EX1016); EX1004, pp.4,12; EX1003, pp.3.

Further, both Tahan and Menahem disclose that SVM feature classification is used as part of malware identifying processes and implement SVMs in tests. EX1002, ¶¶269-272; EX1004, pp.2; EX1003, pp.4. Accordingly, a PHOSITA would have understood that the use of SVM in malware determination disclosed in Tahan-Menahem discloses this claim limitation. EX1002, ¶¶130-149,264-281,365-374.

For the foregoing reasons, a PHOSITA as of the filing date would have found claim 17 to be obvious based on the Tahan-Menahem combination. Accordingly, claim 17 should be found unpatentable over the Tahan-Menahem combination under 35 U.S.C. § 103.

13.Claim 19

The system according to claim 17, wherein the extracting the plurality of predefined character strings further comprises classifying the extracted plurality of predefined character strings according to a type of data extracted from the executable file, and encoding the extracted plurality of predefined character strings for the classifier based on the classification.

Tahan and Menahem both teach generating meta-features of extracted data based on an initial classification with different inducers. EX1002, ¶294; EX1003, pp.9-10,25,28; Ex 1004, p.4-6. Further, Tahan teaches extracting multiple types of features from executable files (e.g., numeric, op-codes, strings, file headers).

EX1002, 284-292; EX1003, pp.7-9. As described regarding claim limitation 17[d], the Tahan-Menahem combination discloses classifying features/predefined character strings and encoding them for a higher-level classifier as meta-features (corresponding to “encoding the extracted plurality of predefined character strings for the classifier based on the classification”). EX1003, pp.4,title; EX1004, pp.4; EX1002, ¶292. The essence of a stacked/meta classifier in Menahem involves encoding of classified data as inputs to further classifiers (see specialist classifier), which a PHOSITA would understand to be the encoding limitation of claim 19. EX1002, ¶¶293-299,377-378; EX1004, 4,8-11,Fig.1.

Further, Tahan discloses that the extracted n-grams can represent binary features or code segments and can be categorized (corresponding to “classifying the extracted...predefined character strings according to a type of data”) according to value spread and other aspects of the n-grams. EX1002, ¶¶288-291,377-378; EX1003, pp.4-11. Likewise, Menahem extracts PE headers and function-based data including numeric, nominal, and Boolean values and strings as input to base classifiers which then classify and apply classifications/encode as meta-features to upstream classifiers. EX1002, ¶¶292-295,377-378; EX1004, pp.4-11. Thus, a PHOSITA would have understood the generation of classified features / predefined character strings for the classifiers of Tahan and Menahem as classifying and encoding predefined character strings as recited in claim 19. EX1002, ¶¶284-301.

Also, a PHOSITA would have understood that Tahan-Menahem's "extracting the plurality of predefined character strings further comprises" the above-discussed encoding and classifying of predefined character strings, or would have been motivated to configure it in such a manner. EX1002, ¶¶130-149,300,377-378.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 19 to be obvious based on the Tahan-Menahem combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 17 from which claim 19 depends. Accordingly, claim 19 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem.

B. Ground 2: Claims 6, 7, 9, 15-16, 18 and 20 are unpatentable under 35 U.S.C. § 103 as obvious based on Tahan in view of Menahem in view of Schultz

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). Petitioner is unaware of any secondary considerations relevant to the '844 patent and a more than adequate written rationale is provided to support the combination of Tahan and Menahem under *KSR*. 550 U.S. at 418.

A PHOSITA as of the filing date would have been motivated to combine certain features from Schultz with Tahan-Menahem. Specifically, a PHOSITA

would have been motivated to improve the classifiers of Tahan-Menahem with the methods of Schultz which enables detection of malware in a larger set of executable files. EX1002, ¶383; EX1005, ¶[0012],[0016]; EX1004, p.9.

Specifically, Schultz teaches practical methods for improving Tahan-Menahem by reducing false positives before taking action, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem.

EX1002, ¶383-384; EX1005, ¶[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 are compatible with those of Schultz. EX1003, pp.31; EX1002, ¶¶381-382; EX1004, pp.12. The above motivation to combine analysis is incorporated by reference into each subsection/limitation below.

1. Claim 6

The computer-implemented method according to claim 1, wherein determining whether the executable file is harmful further comprises preventing execution of the executable file when a determined probability value that the executable file is harmful exceeds a threshold value.

Tahan-Menahem teaches the limitations of claim 1 as described above.

Schultz teaches taking actions based on the improved methods of Tahan-Menahem.

Specifically, Schultz discloses quarantining (thus preventing from execution)

harmful executable files attached to emails based on a rule evaluator and analysis.

EX1005, ¶¶ [0103],[0108],[Abstract]; EX1002, ¶¶386-387.

Schultz also discloses that the analysis used to quarantine email attachments (executable files) involves determining if a probability value that an executable file is harmful exceeds a threshold. EX1002, ¶¶386-389. Schultz states:

A exemplary metric to identify borderline cases, which may be implemented in hexScan or a similar routine is to define a borderline case to be a case where the difference between the probability, or score, that the program is malicious and the probability, or score, it is benign is below a threshold. This threshold may be set based on the policies of the host. For example, in a secure setting, the threshold could be set relatively high, e.g., 20%.

EX1005, ¶[0101].

A PHOSITA would have understood that the probabilities and metrics of Schultz are the same as the probability and thresholds recited in claim 6. EX1002, ¶¶386-391; EX1005, ¶[0101]. Schultz teaches methods of reducing false positives before taking actions like preventing file execution, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem. EX1002, ¶386-394; EX1005, ¶¶[0114]-[0120].

Furthermore, because information on whether an executable file is harmful is used to prevent execution of the executable file, a PHOSITA would have

understood that Tahan-Menahem-Schultz discloses or renders obvious “wherein determining whether the executable file is harmful further comprises” the above-discussed technique. EX1002, ¶¶381-384,386-395.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 6 to be obvious based on the Tahan-Menahem-Schultz combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 1 from which claim 6 depends. Accordingly, claim 6 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem in view of Schultz.

2. Claim 7

The computer-implemented method according to claim 6, wherein the threshold value is set based on predetermined false positive range data.

Tahan-Menahem-Schultz teaches the limitations of claim 6 as noted above. Further regarding claim 7, Schultz teaches selection of detection rates and false positive rates as an inherent trade-off that can be set as an accuracy threshold. EX1005, ¶¶[0116],[0121]; EX1002, ¶398,381-384.

Schultz explains that “The detection rate is the number of malicious binaries correctly classified divided by the total number of malicious programs tested.” EX1005, ¶[0117]. A PHOSITA would have understood a more aggressive approach would improve the detection rate (e.g., and help identify more harmful

executables) at an expense of a worsened false positive rate. Also, a POSITA would have understood that a more aggressive approach would set the above-discussed threshold value for a “probability value that the executable file is harmful” (of claims 6 and 7) lower (e.g., which would help block harmful executables), but also at an expense of a worsened false positive rate. Thus, a PHOSITA would have understood that a false positive rate (or false positive range data) is an underlying factor/consideration in setting a malware detection method. Thus, a PHOSITA would have been motivated to use false positive range data in setting the above-discussed threshold such that “the threshold value is set based on predetermined false positive range data.” EX1002, ¶¶381-384,399-400.

Schultz teaches methods of reducing false positives before taking actions like preventing file execution, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem. EX1002, ¶¶399-400; EX1005, ¶¶[0114]-[0120].

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 7 to be obvious based on the Tahan-Menahem-Schultz combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 1 and the Tahan-Menahem-Schultz combination discloses all elements of claim 6 from which claim 7 depends. Accordingly, claim 7 should be

found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem in view of Schultz.

3. 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.

Tahan-Menahem teaches the limitations of claim 1 as described above.

EX1002, ¶403. Further, 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 by an expert). EX1005, ¶¶[0108]-[0110]; EX1002, ¶¶404-405. 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. EX1002, ¶406; EX1005, ¶¶[0109]-[0113].

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 9 to be obvious based on the Tahan-Menahem-Schultz

combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 1 from which claim 9 depends. Accordingly, claim 9 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem in view of Schultz.

4. Claim 15

The computer-readable storage device according to claim 10, wherein determining whether the executable file is harmful further comprises preventing execution of the executable file when a determined probability value that the executable file is harmful exceeds a threshold value.

Tahan-Menahem teaches the limitations of claim 10 as described above.

Schultz teaches taking actions based on the improved methods of Tahan-Menahem.

Specifically, Schultz discloses quarantining (thus preventing from execution)

harmful executable files attached to emails based on a rule evaluator and analysis.

EX1005, ¶¶ [0103],[0108],[Abstract]; EX1002, ¶¶386-387,408-409.

Schultz also discloses that the analysis used to quarantine email attachments (executable files) involves determining if a probability value that an executable file is harmful exceeds a threshold. EX1002, ¶¶386-389. Schultz states:

A exemplary metric to identify borderline cases, which may be implemented in hexScan or a similar routine is to define a borderline case to be a case where the difference between the probability, or score, that the program is malicious and the probability, or score, it is benign is below a threshold. This threshold may be set based on the policies of

the host. For example, in a secure setting, the threshold could be set relatively high, e.g., 20%.

EX1005, ¶[0101].

A PHOSITA would have understood that the probabilities and metrics of Schultz are the same as the probability and thresholds recited in claim 15. EX1002, ¶¶386-391; EX1005, ¶[0101]. Schultz teaches methods of reducing false positives before taking actions like preventing file execution, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem. EX1002, ¶386-394,408-409; EX1005, ¶¶[0114]-[0120].

Furthermore, because information on whether an executable file is harmful is used to prevent execution of the executable file, a PHOSITA would have understood that Tahan-Menahem-Schultz discloses or renders obvious “wherein determining whether the executable file is harmful further comprises” the above-discussed technique. EX1002, ¶¶381-384,386-395,408-409.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 15 to be obvious based on the Tahan-Menahem-Schultz combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 10 from which claim 15 depends. Accordingly, claim 15 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem in view of Schultz.

5. Claim 16

The computer-readable storage device according to claim 15, wherein the threshold value is set based on predetermined false positive range data.

Tahan-Menahem-Schultz teaches the limitations of claim 15 as noted above.

Further regarding claim 16, Schultz teaches selection of detection rates and false positive rates as an inherent trade-off that can be set as an accuracy threshold.

EX1005, ¶¶[0116],[0121]; EX1002, ¶398,381-384,408-409.

Schultz explains that “The detection rate is the number of malicious binaries correctly classified divided by the total number of malicious programs tested.”

EX1005, ¶[0117]. A PHOSITA would have understood a more aggressive approach would improve the detection rate (e.g., and help identify more harmful executables) at an expense of a worsened false positive rate. Also, a POSITA would have understood that a more aggressive approach would set the above-discussed threshold value for a “probability value that the executable file is harmful” lower (e.g., which would help block harmful executables), but also at an expense of a worsened false positive rate. Thus, a PHOSITA would have understood that a false positive rate (or false positive range data) is an underlying factor/consideration in setting a malware detection method. Thus, a PHOSITA would have been motivated to use false positive range data in setting the above-

discussed threshold such that “the threshold value is set based on predetermined false positive range data.” EX1002, ¶¶381-384,399-400,408-409.

Schultz teaches methods of reducing false positives before taking actions like preventing file execution, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem. EX1002, ¶¶399-400,408-409; EX1005, ¶¶[0114]-[0120].

Therefore, a PHOSITA would have found the limitations recited in claim 16 to be obvious based on the Tahan-Menahem-Schultz combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 10 and the Tahan-Menahem-Schultz combination discloses all elements of claim 15 from which claim 16 depends. Accordingly, claim 16 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem in view of Schultz.

6. Claim 18

The system according to claim 17, wherein the determining further comprises preventing execution of the executable file when a determined probability value that the executable file is harmful exceeds a threshold value, and wherein the threshold value is set based on predetermined false positive range data.

Tahan-Menahem teaches the limitations of claim 17 as described above.

Schultz teaches taking actions based on the improved methods of Tahan-Menahem. Specifically, Schultz discloses quarantining (thus preventing from execution)

harmful executable files attached to emails based on a rule evaluator and analysis.

EX1005, ¶¶ [0103],[0108],[Abstract]; EX1002, ¶¶386-387,411-412.

Schultz also discloses that the analysis used to quarantine email attachments (executable files) involves determining if a probability value that an executable file is harmful exceeds a threshold. EX1002, ¶¶386-389. Schultz states:

A exemplary metric to identify borderline cases, which may be implemented in hexScan or a similar routine is to define a borderline case to be a case where the difference between the probability, or score, that the program is malicious and the probability, or score, it is benign is below a threshold. This threshold may be set based on the policies of the host. For example, in a secure setting, the threshold could be set relatively high, e.g., 20%.

EX1005, ¶[0101].

A PHOSITA would have understood that the probabilities and metrics of Schultz are the same as the probability and thresholds recited in claim 18. EX1002, ¶¶386-391; EX1005, ¶[0101]. Schultz teaches methods of reducing false positives before taking actions like preventing file execution, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem.

EX1002, ¶386-394,411-412; EX1005, ¶¶[0114]-[0120].

Furthermore, because information on whether an executable file is harmful is used to prevent execution of the executable file, a PHOSITA would have

understood that Tahan-Menahem-Schultz discloses or renders obvious “wherein the determining [whether the executable file is harmful] further comprises” the above-discussed technique. EX1002, ¶¶381-384,386-395,411-412.

Schultz teaches selection of detection rates and false positive rates as an inherent trade-off that can be set as an accuracy threshold. EX1005, ¶¶[0116],[0121]; EX1002, ¶398,381-384,411-412. Schultz explains that “The detection rate is the number of malicious binaries correctly classified divided by the total number of malicious programs tested.” EX1005, ¶[0117]. A PHOSITA would have understood a more aggressive approach would improve the detection rate (e.g., and help identify more harmful executables) at an expense of a worsened false positive rate. Also, a POSITA would have understood that a more aggressive approach would set the above-discussed threshold value for a “probability value that the executable file is harmful” lower (e.g., which would help block harmful executables), but also at an expense of a worsened false positive rate. Thus, a PHOSITA would have understood that a false positive rate (or false positive range data) is an underlying factor/consideration in setting a malware detection method. Thus, a PHOSITA would have been motivated to use false positive range data in setting the above-discussed threshold such that “the threshold value is set based on predetermined false positive range data.” EX1002, ¶381-384,399-400,411-412.

Schultz teaches methods of reducing false positives before taking actions like preventing file execution, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem. EX1002, ¶¶399-400,408-409; EX1005, ¶¶[0114]-[0120].

For these two reasons, a PHOSITA as of the filing date would have found the limitations recited in claim 18 to be obvious based on the Tahan-Menahem-Schultz combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 17 from which claim 18 depends. Accordingly, claim 18 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem in view of Schultz.

7. Claim 20

The system according to claim 17, wherein determining whether the executable file is harmful further comprises preventing execution of the executable file when a determined probability value that the executable file is harmful exceeds a threshold value.

Tahan-Menahem teaches the limitations of claim 17 as described above. Schultz teaches taking actions based on the improved methods of Tahan-Menahem. Specifically, Schultz discloses quarantining (thus preventing from execution) harmful executable files attached to emails based on a rule evaluator and analysis. EX1005, ¶¶ [0103],[0108],[Abstract]; EX1002, ¶¶386-387,411-412.

Schultz also discloses that the analysis used to quarantine email attachments (executable files) involves determining if a probability value that an executable file is harmful exceeds a threshold. EX1002, ¶¶386-389. Schultz states:

A exemplary metric to identify borderline cases, which may be implemented in hexScan or a similar routine is to define a borderline case to be a case where the difference between the probability, or score, that the program is malicious and the probability, or score, it is benign is below a threshold. This threshold may be set based on the policies of the host. For example, in a secure setting, the threshold could be set relatively high, e.g., 20%.

EX1005, ¶[0101].

A PHOSITA would have understood that the probabilities and metrics of Schultz are the same as the probability and thresholds recited in claim 20. EX1002, ¶¶386-391; EX1005, ¶[0101]. Schultz teaches methods of reducing false positives before taking actions like preventing file execution, which a PHOSITA would have recognized as useful in deploying the detection methods of Tahan-Menahem.

EX1002, ¶386-394,411-412; EX1005, ¶¶[0114]-[0120].

Furthermore, because information on whether an executable file is harmful is used to prevent execution of the executable file, a PHOSITA would have understood that Tahan-Menahem-Schultz discloses or renders obvious “wherein

determining whether the executable file is harmful further comprises” the above-discussed technique. EX1002, ¶¶381-384,386-395,411-412.

Therefore, a PHOSITA as of the filing date would have found the limitations recited in claim 20 to be obvious based on the Tahan-Menahem-Schultz combination. As shown, the Tahan-Menahem combination discloses all elements recited in claim 17 from which claim 20 depends. Accordingly, claim 20 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Tahan in view of Menahem in view of Schultz.

VII. NO BASIS FOR DISCRETIONARY DENIAL

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

There is parallel court litigation, so *Fintiv* factors are addressed preliminarily. First, no stay has been entered in the parallel litigation. Second, trial is currently set for August 19, 2024, so there is time to complete inter parte review. Third, it is early in the parallel litigation as fact discovery has not started. Fourth, Petitioner has identified the Tahan, Menahem and Schultz prior art in Defendant’s invalidity contentions filed in the parallel litigation. 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 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 and Schultz were not cited, individually or in combination, in any of the 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 and Schultz are materially different from the combination of references asserted in Office communications. Thus, the individual references of Tahan, Menahem 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.

Even if the Board finds that the “same or substantially the same” art or arguments were presented to the Office, discretionary denial under § 325(d) is still not proper because the Office made a “material error” in not considering the Tahan, Menahem and Schultz references.

Regarding *Becton* factors (c) (evaluation of art during examination), none of Tahan, Menahem 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* factor (e) (examiner error) and (f) (other facts), this petition specifically points out and explains how Tahan, Menahem and Schultz disclose all limitations of the claims.

VIII. CONCLUSION

Inter partes review of claims 1-20 of U.S. Patent No. 10,599,844 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", is 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: March 3, 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. 10,599,844**, Petitioner's power of attorney, this petition for *inter partes* review, and all associated exhibits were served in their entireties on March 3, 2023, on the following party via Express Mail: to the Patent Owner by serving to the correspondence address of record for the '844 patent:

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

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

Robert M. Hansen, Reg. No. 43,656

Attorney for Petitioner

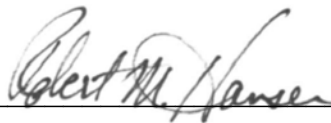
Date: March 3, 2023

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 **13,917** words, excluding the table of contents, table of authorities, exhibit list, mandatory disclosures, signature block, certificates and attached exhibits of the Petition.

The Marbury Law Group, PLLC

A handwritten signature in dark ink, appearing to read "Robert M. Hansen", is written over a horizontal line.

Robert M. Hansen, Reg. No. 43,656
Attorney for Petitioner

Date: March 3, 2023

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

Tel: 703-391-2900