

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Trend Micro, Inc.,

Petitioner

v.

Open Text Inc.,

Patent Owner

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

PATENT OWNER'S PRELIMINARY RESPONSE

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

	PAGE
I. INTRODUCTION	1
II. THE BOARD SHOULD EXERCISE ITS DISCRETION TO DENY INSTITUTION UNDER 35 U.S.C. § 314(A).....	3
A. <i>Fintiv</i> Factor 1: The District Court Has Not Granted And Will Not Grant A Stay.....	6
B. <i>Fintiv</i> Factor 2: A District Court Trial Is Expected Before The Projected Statutory Deadline.....	11
C. <i>Fintiv</i> Factor 3: The Parties Will Have Invested Significant Resources In The District Court Proceeding.....	13
D. <i>Fintiv</i> Factor 4: The Issues Overlap Significantly.	15
E. <i>Fintiv</i> Factor 5: Both Parties Are Parties In The Parallel Litigation	16
F. <i>Fintiv</i> Factor 6: The Petition Does Not Present Compelling Evidence Of Un-patentability.....	18
III. THE PETITION IS NOT REASONABLY LIKELY TO SUCCEED.	19
A. Background	22
1. The ’869 Patent	24
2. The Board’s Rejection of Petitioner’s IPR Petition Against the Parent ’844 Patent Based on Overlapping Art.....	29
3. The Board’s Rejection of Petitioner’s Co-Defendant’s PGR Petition Against the ’869 Patent.....	30
4. The Board’s Rejection of Petitioner’s Co-Defendant’s IPR	

Patent Owner’s Preliminary Response
to Petition for IPR2023-01380
Patent No. 11,409,869

Petition Against the Parent ’844 Patent.	31
5. Claim Construction	32
6. The Level of Ordinary Skill	34
B. Overview of the cited art.	34
1. Overview of Tahan.....	34
2. Overview of Menahem	39
3. Overview of Forman	41
C. Legal Standards	43
D. Ground 1: The combination of Tahan, Menahem, and Forman did not render claims 1-4, 8, 10-25, 29, 32-39, 43, 46-53, and 57 obvious.	43
1. The proposed combination fails to teach turning features on or off at the CLASSIFICATION STAGE.	45
i. Tahan	45
ii. Combination with Menahem	47
iii. Combination with Forman.....	48
2. The proposed combination fails to teach turning features on or off based on the values of static data points extracted from the executable being classified.....	51
i. Tahan	52
ii. Combination with Menahem	53

iii.	Combination with Forman.....	54
3.	The proposed combination fails to teach evaluation of the recited feature vectors using a “support vector machine” or any other “machine learning model.”	56
i.	Tahan	57
ii.	Combination with Menahem	58
iii.	Combination with Forman.....	60
E.	Ground 2: The combination of Tahan, Menahem, and Schultz did not render dependent claims 5-7, 9, 26-28, 30-31, 40-42, 44-45, 54-56, and 58-59 obvious.....	61
IV.	THERE ARE OBJECTIVE INDICIA OF NON-OBVIOUSNESS.....	62
V.	CONCLUSION.....	64

TABLE OF AUTHORITIES

Cases:

<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020)	<i>passim</i>
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 15 (PTAB May 13, 2020) (<i>informative</i>)	12
<i>Commscope Techs. v. Dali Wireless, Inc.</i> , IPR 2022-01242	20
<i>Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015)	44
<i>Eclinical Works, LLC v. Decapolis</i> LLC, IPR2022-00229, Paper 10 (PTAB Apr. 13, 2022)	13
<i>Everstar Merch. Co. v. Willis Elec. Co.</i> , No. 2021-1882, 2022 WL 1089909 (Fed. Cir. Apr. 12, 2022)	50
<i>Google LLC v. EcoFactor, Inc.</i> , IPR2021-08694, Paper 9 (PTAB July 22, 2021)	17
<i>Harmonic Inc. v. Avid Tech., Inc.</i> , 815 F.3d 1356 (Fed. Cir. 2016)	44, 50
<i>In re Huang</i> , 100 F.3d 135 (Fed. Cir. 1996)	63
<i>Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA, slip op. (W.D. Tex. Dec. 22, 2022)	8
<i>Leo Pharm. Prods., Ltd. v. Rea</i> , 726 F.3d 1346 (Fed. Cir. 2013)	63
<i>Lumenis Be Ltd. v. BTL Healthcare Techs. A.S.</i> , IPR2021-01280, Paper 8 (PTAB Mar. 9, 2022)	44

Patent Owner’s Preliminary Response
to Petition for IPR2023-01380
Patent No. 11,409,869

<i>Luxshare Precision Industry v. Amphenol Corp.,</i> IPR2022-00132, Paper 10 (PTAB May 3, 2022)	11
<i>mCom IP, LLC v. Cisco Sys., Inc.,</i> No. 6:22-00261-ADA, slip op. (W.D. Tex. Oct. 20, 2022).....	8
<i>Motorola Mobility LLC v. Ironworks Patents, LLC,</i> IPR2021-00421, Paper 11 (PTAB July 22, 2021)	12
<i>NHK Spring Co., Ltd. v. Intri-Plex Techs., Inc.,</i> IPR2018-00752, Paper 8 (PTAB Sept. 12, 2018) (<i>precedential</i>).....	13
<i>Nintendo Co. v. Gamevice, Inc.,</i> IPR2020-01197, Paper 13 (PTAB Jan. 12, 2021)	11
<i>Nokia of Am. Corp. v. Godo Kaisha IP Bridge,</i> IPR2022-00755, Paper 9 (PTAB Nov. 2, 2022).....	18
<i>Regeneron Pharms. Inc. v. Novartis Pharma AG,</i> IPR2020-01317, Paper 15 (PTAB Jan. 15, 2021)	15
<i>Robert Bosch Tool Corp. v. SD3, LLC,</i> IPR2016-01751, Paper 15 (PTAB Mar. 22, 2017)	63
<i>Samsung Elecs. Co. v. California Institute of Tech.,</i> IPR2023-00131, Paper 10 (PTAB May 4, 2023)	13
<i>Samsung Elecs. Co. v. Clear Imaging Research LLC,</i> IPR2020-01399, Paper 13 (PTAB Feb. 3, 2021).....	11, 12
<i>Sonrai Memory Ltd. v. Kingston Tech. Co., Inc.,</i> No. 6:21-cv-1284-ADA, slip op. (W.D. Tex. Oct. 18, 2022).....	8
<i>Sophos Ltd. et al. v. Open Text,</i> PGR-2023-0031, Paper 7 (PTAB Nov. 29, 2023).....	31, 32, 46
<i>Sophos Ltd. et al. v. Webroot,</i> IPR2023-00528, Paper 14 (PTAB August 14, 2023)	32, 33

Patent Owner’s Preliminary Response
to Petition for IPR2023-01380
Patent No. 11,409,869

<i>Sotera v. Masimo Corp.</i> , IPR2020-01019 (PTAB Dec. 1, 2020)	17
<i>Trend Micro Inc.</i> , IPR2023-00662, Paper 8 (PTAB Oct. 13, 2023)	<i>passim</i>
<i>Trend Micro, Inc. v. Webroot Inc.</i> , IPR2023-00662, Paper 1 (PTAB Mar. 3, 2023)	19, 30, 54, 59
<i>Twinstrand Biosciences, Inc. v. Guardant Health, Inc.</i> , IPR2022-01115, Paper 14 (PTAB Dec. 30, 2022)	62
<i>Vizio, Inc. v. Maxell, Ltd.</i> , IPR2022-01459, Paper 7 (PTAB Apr. 11, 2023)	18
Statutes	
35 U.S.C. § 312(a)(3)	44
35 U.S.C. § 314(a)	2, 4, 19
35 U.S.C. § 325(d)	2
Other Authorities	
37 C.F.R. § 42.22(a)(2)	44
37 C.F.R. § 42.100(b)	33
Director’s Memo, <i>Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation</i> (June 21, 2022)	4, 5, 12, 20
H.R. Rep. No. 112-98, pt. 1 (2011), 2011 U.S.C.C.A.N. 67	1

EXHIBIT LIST

Exhibit	Description
2001	Declaration Of Sam Malek, Ph.D.
2002	Curriculum Vitae of Sam Malek, Ph.D.
2003	<i>Webroot, Inc., Open Text Inc., v. AO Kaspersky Lab et al.</i> , Case No. 6:22-cv-00243-ADA-DTG, Order Granting Amended Scheduling Order, Dkt. 521 (W.D. Tex., Dec. 7, 2023)
2004	<i>Sonrai Memory Ltd. v. Kingston Tech. Co. and Kingston Tech. Corp.</i> , Case No. 6:21-cv-1284-ADA, Order Denying Kingston Tech. Co. and Kingston Tech. Corp.'s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 94 (W.D. Tex., Oct. 18, 2022)
2005	<i>mCom IP, LLC v. Cisco Systems, Inc.</i> , Case No. 6:22-cv-00261-ADA, Order Denying Defendant's Motion to Stay Pending <i>Inter Partes</i> Review, Dkt. 42 (W.D. Tex., Oct. 20, 2022)
2006	<i>Intellectual Ventures I LLC et al. v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA, Order Denying Hewlett Packard Enterprise Company's Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 104 (W.D. Tex., Dec. 22, 2022)
2007	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Claim Construction Order, Dkt. 236 (W.D. Tex. Mar. 16, 2023)
2008	Dani Kass, "Catching Up On Patent Litigation With Judge Albright" (Law360 Mar. 14, 2023); https://www.law360.com/pulse/articles/1582438/print?section=pulse/courts
2009	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Claim Construction Order, Dkt.

Patent Owner's Preliminary Response
to Petition for IPR2023-01380
Patent No. 11,409,869

Exhibit	Description
	391 (W.D. Tex. July 25, 2023)
2010	Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al., Case No. 6:22-cv-243-ADA-DTG, Dec. 9, 2022 Transcript of Motions Hearing, Dkt. 133 (W.D. Tex. Dec. 15, 2022) (excerpted)
2011	<i>Webroot, Inc. and Open Text Inc., v. Trend Micro, Inc.</i> , Case No. 6:22-cv-00239-ADA, Complaint for Patent Infringement, Dkt. 1 (W.D. Tex., Mar. 4, 2022)
2012	<i>Webroot, Inc. and Open Text Inc., v. Ao Kaspersky Lab</i> , Case No. 6:22-cv-00243-ADA-DTG, Defendant's Preliminary Invalidity Contentions Appendix C (W.D. Tex., Sept. 14, 2022)
2013	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-00240-ADA-DTG, Defendant Sophos Ltd.'s Final Invalidity Contentions (W.D. Tex., Nov. 1, 2023) (excerpted)
2014	<i>Webroot, Inc. and Open Text Inc., v. Trend Micro Inc.</i> , Case No. 6:22-cv-00239-ADA-DTG, Defendant's Final Invalidity Contentions (W.D. Tex., Nov. 1, 2023) (excerpted)

Open Text Inc. (“Patent Owner” or “Open Text”) submits this Preliminary Response to the Petition of Trend Micro, Inc. (“Petitioner”) seeking *inter partes* review (“IPR”) of U.S. Patent No. 11,409,869 (“the ’869 patent”).

I. INTRODUCTION

The Board should deny institution for two reasons: (1) discretionary denial under *Fintiv* is appropriate because the ’869 patent is part of an exceptionally complex parallel litigation that is scheduled for trial months before a final written decision on this petition would be due, and (2) the petition is not reasonably likely to succeed because it relies on the same art and arguments the Board already rejected when it denied an earlier petition for IPR of the ’869 patent’s parent (U.S. Patent No. 10,599,844; the “’844 patent”). See *Trend Micro Inc.*, IPR2023-00662, Paper 8 (PTAB Oct. 13, 2023) (“’844 Trend Denial”).

First, the Board should deny institution under *Fintiv* because instituting this IPR would undermine Congress’s stated goals in creating IPR proceedings: to provide “quick and cost effective alternatives to litigations” and “limit unnecessary and counterproductive litigation costs.” H.R. Rep. No. 112-98, pt. 1, at 40 (2011), 2011 U.S.C.C.A.N. 67, 69. An IPR here would be neither quicker than, nor an alternative to, ongoing district court litigation. The Western District of Texas (“WDTX”) is scheduled to proceed to trial on the ’869 patent almost four months

before a final written decision on the petition would be due—so an IPR is not quicker. Moreover, the district court has indicated it will not issue a stay when, as here, it can resolve the same issues prior to the Board—so an IPR is not an alternative to litigation. Finally, other defendants have asserted the invalidity grounds raised in the petition, which the district court must address regardless of whether the Board institutes an IPR. Therefore, instituting an IPR would generate the unnecessary and counterproductive litigation costs that Congress sought to avoid.

Second, the Board should deny institution under 35 U.S.C. §§ 314(a) and 325(d) because Petitioner is unlikely to win on the merits. The cited art does not teach or suggest (1) turning features on or off at the classification stage, (2) doing so based on the values of static data points being within a predetermined range, or (3) evaluating the resulting feature vector using “support vector processing.”

Indeed, The Board has *already agreed* with the latter two arguments—citing both *sua sponte* among its reasons for rejecting Petitioner's earlier IPR petition against the '869 patent's parent (the '844 patent). That earlier petition pressed the same primary and secondary references as here (Tahan and Menahem), against substantially the same claim language, using substantially the same arguments. *See '844 Trend Denial*. The addition here of Forman as a tertiary reference does not remedy the shortcomings the Board identified in Petitioner's earlier petition.

II. THE BOARD SHOULD EXERCISE ITS DISCRETION TO DENY INSTITUTION UNDER 35 U.S.C. § 314(a).

The '869 patent is one of dozens of patents asserted in claims and counterclaims among four different parties in a series of cases pending in the Western District of Texas. The district court has already issued its claim constructions and is on track to adjudicate the same prior art and invalidity grounds raised in the petition months before the Board would issue a final written decision. Instituting a parallel IPR proceeding that lags far behind the district court would needlessly increase the cost of resolving the underlying dispute. The Board should therefore exercise its discretionary authority to deny institution under 35 U.S.C. § 314(a).

Because Congress's goal in enacting post-grant proceedings like IPRs was "to ... limit unnecessary and counterproductive litigation costs," Director Vidal has instructed the Board not to institute IPRs that would "increase, rather than limit, litigation costs." Director's Memo, *Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation* ("Guidance") at 1 (June 21, 2022).

Instituting this IPR would cause just such an increase. The reasons are easy to see: a district court stay is out of the question, and the district court will consider the

very same arguments presented in the petition. Accordingly, the Board and the district court will labor to decide the very same issues with the district court well ahead of—and scheduled to reach a final decision months before—the Board. The year-long parallel tracks of two adjudicatory bodies will “increase, rather than limit, litigation costs,” create “potential conflict between the ... (PTAB) and district court proceedings,” and generally increase “the potential for inefficiency and gamesmanship.” Guidance at 1. This is precisely the situation that Director Vidal sought to avoid when she instructed the Board to take “a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review.” *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 6 (PTAB Mar. 20, 2020).

Confirming this conclusion, every one of the *Fintiv* factors weighs for discretionary denial:

(1) A stay of the co-pending district court litigation—involving 27 patents and three defendants—is highly unlikely. Petitioner cannot avoid a district court trial based on an institution of this Petition, and trial on the Petition will not narrow the issues on which discovery and trial must proceed. Indeed, the '869 patent is currently asserted against all three defendants in a now-consolidated case. Moreover, none of Petitioner's codefendants have agreed not to litigate the validity of the '869 patent

in the district court if the Board institutes this IPR.

(2) Trial in the district court is scheduled to occur nearly four months before a final written decision, and the district court case is on track for that trial date.

(3) The parties and district court have already invested heavily in carrying the parallel litigation past claim construction and well into discovery.

(4) The issues raised in the petition substantially overlap with those raised in the district court, including Trend asserting in the district court the same primary reference and three of the four total references., and Petitioner has not offered *any* stipulation that it will not simultaneously pursue in the district court *the same arguments* based on *the same grounds* asserted in its petition during the pendency of any IPR instituted from that petition, irrespective of the Board's institution decision.

(5) Petitioner will get its chance to make its arguments before the district court because Petitioner is a defendant in the parallel litigation.

(6) As shown below, the petition's grounds of unpatentability are weak.

Thus, every *Fintiv* factor here weighs for discretionary denial. The district court is well on its way to an efficient resolution of the disputes between Patent Owner, Petitioner, and Petitioner's codefendants. Both the goals of the AIA and the general welfare of the patent system would be best served by denying institution.

A. *Fintiv* Factor 1: The District Court Has Not Granted And Will Not Grant A Stay.

The first *Fintiv* factor asks “whether [1] the court granted a stay or [2] evidence exists that one may be granted if a proceeding is instituted.” *Fintiv*, IPR2020-00019, Paper 11 at 6.¹ Here, the undisputed answer to both questions is **no**: no stay has been requested or ordered, and Petitioner has identified no evidence that one will (or even might) be granted if a proceeding is instituted. To the contrary, all evidence indicates that a stay is out of the question.

¹ The decision in *Fintiv* addresses various scenarios that may indicate whether a stay may be granted. *Fintiv*, IPR2020-00019, Paper 11 at 6-9. However, nothing in *Fintiv* suggests that these are the only scenarios that a panel may consider. In particular, *Fintiv* does not hold that an actual denial of a stay is the only consideration relevant to this factor. Rather, *Fintiv* contemplates that any type evidence may be relevant. For example, *Fintiv* holds that one way to “support the arguments that ... a stay may be less likely, and instituting would lead to duplicative costs” is to show the “work completed by the parties and court in the parallel proceeding.” *Fintiv* IPR2020-00019, Paper 11 at 10.

The district court has stated expressly that it considers issue simplification “the most important factor” in deciding whether to stay patent litigation²—and here, a stay would not simplify anything. The validity of the ’869 patent will be litigated in the district court, whether this petition is instituted or not, because the patent is asserted against two other defendants, none of whom have promised to forego their validity challenges in the district court based on this institution decision.

² See, e.g., EX-2004, *Sonrai Memory Ltd. v. Kingston Tech. Co., Inc.*, No. 6:21-cv-1284-ADA, slip op. at *4-5 (W.D. Tex. Oct. 18, 2022) (denying a stay following IPR institution because issue simplification was the “most important factor” and the Court had already held a *Markman* hearing and “expended significant resources”); EX-2005, *mCom IP, LLC v. Cisco Sys., Inc.*, No. 6:22-00261-ADA, slip op. at *5 (W.D. Tex. Oct. 20, 2022) (finding “simplification is not likely” where the IPR did not challenge all of the claims asserted in the district court, since the other claims would still require a trial); EX-2006, *Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.*, No. 6:21-cv-00596-ADA, slip op. at *7 (W.D. Tex. Dec. 22, 2022)(denying a stay following IPR institution , as even IPR estoppel would not simplify the invalidity case, given the defendants’ reliance on prior art systems).

Moreover, the validity of the '869 patent will be litigated in the district court alongside issues regarding 26 *other* patents asserted by and against Petitioner and *two other* defendants—issues that the district court is highly unlikely to delay deciding in favor of letting the Board proceed with *this* IPR. The case against Petitioner alone involves seventeen patents—eleven asserted by Patent Owner and six asserted in counterclaims by Petitioner.

The table below lists the patents, parties, and IPR petitions involved in this highly complex litigation:

**Table 1. Patent Owner's Asserted Patents
Against Each WDTX Defendant and Resulting Review Petitions**

Patent	Kaspersky	Sophos	Trend Micro
11,409,869	IPR2023-01160	PGR2023-00031 (Denied)	IPR2023-01380
8,201,243	IPR2024-00292	IPR2023-00731 (Denied)	IPR2023-01390
		IPR2024-00252	
8,719,932	PR2024-00297	IPR2023-00732 (Denied)	IPR2023-01391
		IPR2024-00253	
8,181,244	Asserted; no IPR	Asserted; no IPR	IPR2023-01392
8,418,250	IPR2023-01334	IPR2023-0699	IPR2023-00655 (Denied)
			IPR2024-0106
8,726,389	IPR2024-0035	IPR2023-00633	IPR2023-01459
10,599,844	Asserted; no IPR	IPR2023-00528 (Denied)	IPR2023-00662 (Denied)

Patent Owner's Preliminary Response
to Petition for IPR2023-01380
Patent No. 11,409,869

Patent	Kaspersky	Sophos	Trend Micro
11,409,869	IPR2023-01160	PGR2023-00031 (Denied)	IPR2023-01380
9,413,721	N/A	IPR2023-00491 (Denied)	N/A
9,578,045	IPR2023-01199	Asserted; no IPR	IPR2023-00677 (Denied)
10,257,224	IPR2023-01011	Asserted; no IPR	IPR2023-00657 (Denied)
10,284,591	IPR2023-00895 (Denied)	Asserted; no IPR	IPR2023-00692 (Denied)
8,856,505	Asserted; no IPR	Asserted; no IPR	Asserted; no IPR
8,763,123	N/A	Asserted; no IPR	N/A
8,438,386	N/A	N/A	N/A
10,025,928	N/A	N/A	N/A

As U.S. Magistrate Judge Gilliland noted during a December 9, 2022 hearing, this is “a phenomenal case that I’m hoping one day maybe people will write songs about because it is about as complicated as you’re going to get with patent cases.” (EX-2010 52.) Consequently, should the Board institute trial on this Petition, the institution will not cause the district court to stay this complex litigation.

Second, the district court has been clear in public comments that it will not stay cases where, as here, the court can resolve the case as quickly as the Board can:

[T]here’s no unfairness to either side by not staying a case for the PTAB when I can try the case as quickly as they can resolve it.... And so far,

I think that's proven to be true.

(EX-2008 3.) Indeed, as commentators have observed, "Judge Albright almost never stays proceedings to let the PTAB make its decision first." (*Id.* 2.)

Judge Albright's public comments apply squarely to the situation here. Trial in the district court case against Petitioner is currently scheduled for November 2024. (EX-2003.) The Board's final written decision if IPR is instituted would not be expected until late March 2025. That Judge Albright has made public comments indicating that his practice is to deny a stay in precisely the circumstances presented here is further evidence that he would not stay the WDTX case in response to institution of an IPR here. (EX-2005 - Court Order Deny Stay Dkt 42 (2022-10-20) 22-cv-00261).

Even where no stay has been requested, the Board has found Factor 1 to favor discretionary denial where (as here) "Petitioner present[ed] no evidence that it has requested a stay [] or that a stay is likely." *Luxshare Precision Industry v. Amphenol Corp.*, IPR2022-00132, Paper 10, at 9 (PTAB May 3, 2022). And such a decision is particularly appropriate where the circumstances show that "a stay ... is unlikely." *Nintendo Co. v. Gamevice, Inc.*, IPR2020-01197, Paper 13, at 11-12 (PTAB Jan. 12, 2021); *Samsung Elecs. Co. v. Clear Imaging Research LLC*, IPR2020-01399, Paper 13 at 11-12 (PTAB Feb. 3, 2021) (holding Factor 1 may weigh for denial where there

is evidence that the district court is unwilling “to avoid duplicative efforts and await the PTAB’s final resolution”); *Motorola Mobility LLC v. Ironworks Patents, LLC*, IPR2021-00421, Paper 11 at 7-8 (PTAB July 22, 2021).

In sum, the district court has not granted a stay and there is no evidence suggesting that the district court would grant a stay of a multi-patent, multi-defendant case so that the Board can reach a decision months *after* the current trial date on *just one* patent. Factor 1 weights in favor of discretionary denial. *Samsung*, IPR2020-01399, Paper 13 at 11-12; *Motorola*, IPR 2021-00421, Paper 11 at 7-8.

B. *Fintiv* Factor 2: A District Court Trial Is Expected Before The Projected Statutory Deadline

Factor 2—the implications of the district court and IPR schedules—also weighs for denial. “[A] parallel proceeding in an advanced state implicates considerations of efficiency and fairness, which can serve as an independent reason to apply discretion to deny institution.” *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 15 at 11-12, 17 (PTAB May 13, 2020) (*informative*); Guidance at 1.

Here, an IPR would extend months past the parallel district court litigation. Trial is currently scheduled for November 2024. (EX-2003.) Should the Board institute, its final written decision deadline would be in late March 2025, nearly four months after the currently scheduled trial against Petitioner.

Thus, if the Board institutes IPR here, the IPR proceedings would not produce a result before the trial. Instead, the parties would waste a year expending precisely the “duplicative,” “unnecessary[,] and counterproductive litigation costs” that the AIA was designed to avoid. *Fintiv*, IPR2020-00019, Paper 11 at 5 (citing *Magellan Midstream Partners L.P v. Sunoco Partners Mktg. & Terms. L.P.*, IPR2019-01445, Paper 12 at 10 (Jan. 22, 2020)); *see id.*, Paper 15 at 17 (discretionarily denying institution where district court trial was scheduled two months before the final decision deadline); *NHK Spring Co., Ltd. v. Intri-Plex Techs., Inc.*, IPR2018-00752, Paper 8 at 20 (PTAB Sept. 12, 2018) (*precedential*) (finding that institution of IPR when the district court trial would resolve validity before the PTAB “would not be consistent with an objective of the AIA...to provide an effective and efficient alternative to district court litigation”); *Samsung Elecs. Co. v. California Institute of Tech.*, IPR2023-00131, Paper 10 at 15, 24 (PTAB May 4, 2023) (discretionarily denying institution where “[r]egardless of whether we consider the scheduled trial date or the trial date expected based on time-to-trial statistics, the trial in the underlying litigation would commence several months before the expected date of our final written decision”); *Eclinical Works, LLC v. Decapolis LLC*, IPR2022-00229, Paper 10 at 9 (PTAB Apr. 13, 2022) (discretionarily denying institution where WDTX trial was scheduled “roughly one to two months before any final

decision would have been due”).

Because the district court will resolve validity before the Board reaches its final written decision, the second *Fintiv* factor also favors discretionary denial.

C. *Fintiv* Factor 3: The Parties Will Have Invested Significant Resources In The District Court Proceeding.

Factor 3—the resources invested in the parallel district court proceeding—favors denial “if, at the time of the institution decision, the district court has issued substantive orders related to the patent.” *Fintiv*, IPR2020-00019, Paper 11 at 9-10; *see also id.* at n.16 (collecting cases). *Fintiv* holds specifically that “claim construction orders may indicate that the court and parties have invested sufficient time in the parallel proceeding to favor denial.” *Id.* at 10.

The parties have invested significant resources in the parallel litigation over the last twenty-one months. For example, the parties have already exchanged their final infringement and invalidity contentions, the district court has held multiple *Markman* hearings, and it issued its claim construction order for the ’869 Patent on July 25, 2023. (EX-2009.)

Fact discovery is substantially underway. (EX-2003.) Petitioner has produced over 4,000 documents and Patent Owner has made approximately 48 TB of source code available. The Court has also held multiple discovery hearings, and Patent

Owner has noticed over 98 depositions across the litigations.

Furthermore, Petitioner's and Patent Owner's experts have already invested many hours preparing for their reports by reviewing the patent, prior art, and the defendants' invalidity contentions, as well as traveling to review source code. *See Regeneron Pharms. Inc. v. Novartis Pharma AG*, IPR2020-01317, Paper 15 at 17 (PTAB Jan. 15, 2021) (denying institution where the parties had expended considerable resources in claim construction, substantial fact discovery, and preparation for expert reports). Institution will not reduce the substantial expenses in the related litigation, as the district court trial will be held months before the Board reaches its final written decision.

Petitioner's delay in filing this petition further weighs for denial. Petitioner waited more than eight months after receiving Patent Owner's initial infringement contentions before filing its petition. During this time, Petitioner knew that the parties were investing in a rapidly proceeding litigation. Petitioner offers no explanation for its delay, which weighs for denial. *Fintiv*, IPR2020-00019, Paper 11 at 11-12; *Regeneron*, IPR2020-01317, Paper 15 at 17 (finding that even where the petition was filed "less than a month after learning of the complaints," the investment by the parties in the parallel proceeding "outweighs the effort expended so far in this proceeding").

D. *Fintiv* Factor 4: The Issues Overlap Significantly.

Factor 4—the degree of overlap between the issues in the petition and those before the district court—also strongly supports denial.

Regarding this fourth factor, *Fintiv* explains that “concerns of inefficiency and the possibility of conflicting decisions [are] particularly strong” where “the Board [is] presented with substantially identical prior art arguments...at issue in district court.” *Fintiv*, IPR2020-00019, Paper 11 at 12. “Accordingly, if the petition includes the same or substantially the same claims, grounds, arguments, and evidence as presented in the parallel proceeding, this fact has favored denial.” *Id.*

Here, Petitioner has not offered *any* stipulation that it will not simultaneously pursue in the district court *the same arguments* based on *almost the exact same grounds* asserted in its petition during the pendency of any IPR. Indeed, Petitioner raises the same primary reference and three of the four total references in its district court invalidity contentions.³ Even if Petitioner had offered a stipulation, such a stipulation would carry no weight because it does not bar the other defendants in the

³ Petitioner asserts Tahan, Menahem, and Schultz in its invalidity contentions. (EX-2014.)

parallel litigation from litigating substantially the same invalidity challenges before the district court. Petitioner's co-defendant Sophos cites the same primary prior art reference (Tahan) in the litigation that Petitioner asserts here in the petition, and all of the co-defendants cite Schultz.⁴ Instituting an IPR would both multiply the burden on Patent Owner and create the possibility of inconsistent decisions between the district court and the Board.

Because Petitioner makes no effort to “mitigate[] any concerns of duplicative efforts between the district court and the Board,” or “of potentially conflicting decisions,” Factor 4 weighs heavily in favor of discretionary denial. *Sotera v. Masimo Corp.*, IPR2020-01019 (PTAB Dec. 1, 2020), Paper 12 at 19; *Google LLC v. EcoFactor, Inc.*, IPR2021-08694, Paper 9 at 12-14 (PTAB July 22, 2021).

E. *Fintiv* Factor 5: Both Parties Are Parties In The Parallel Litigation

Factor 5—the overlap of the parties between the petition and the parallel litigation—suggests denial because Petitioner is a defendant in the parallel litigation and, as such, will have a full, fair opportunity to litigate its invalidity contentions.

⁴ Codefendants Sophos cites Tahan, Menahem, and Schultz. and Kaspersky cites Schultz. (EX-2013; EX-2012.)

Under *Fintiv*, “[i]f a petitioner is unrelated to a defendant in an earlier court proceeding, the Board has weighed this fact against exercising discretion to deny institution.” *Fintiv*, IPR2020-00019, Paper 11 at 13-14. That is not the case here because Petitioner is a defendant in the litigation. Accordingly, the circumstances raise no fairness concerns that denial would deprive Petitioner of its “day in court.” See *Vizio, Inc. v. Maxell, Ltd.*, IPR2022-01459, Paper 7 at 21-22 (PTAB Apr. 11, 2023) (“Because Petitioner is the litigation defendant, factor 5 weighs in favor of discretionary denial”); *Nokia of Am. Corp. v. Godo Kaisha IP Bridge*, IPR2022-00755, Paper 9 at 13 (PTAB Nov. 2, 2022) (holding that Factor 5 weighed for discretionary denial where Petitioner was defendant in parallel proceeding).

That another defendant is litigating substantially the same petitioned issues in the district court—specifically, obviousness based primarily on Tahan—further supports denial. *Fintiv* holds that, “[e]ven when a petitioner is unrelated to a defendant, however, if the issues are the same as, or substantially similar to, those ... about to be litigated, or other circumstances weigh against redoing the work of another tribunal, the Board may, nonetheless, exercise the authority to deny institution.” *Fintiv*, IPR2020-00019, Paper 11 at 14. Because of this consideration, *Fintiv* goes on to require that “[a]n unrelated petitioner ... discuss why addressing the same or substantially the same issues would not be duplicative of the [other] case

even if the petition is brought by a different party.” *Id.*

Petitioner offers no such explanation here. That other defendants are currently litigating the same obviousness issue based on the same primary reference raises serious concerns of duplicative work by adjudicators. Any public-policy concerns regarding patent validity will be thoroughly addressed by the parallel litigation. Consequently, this factor too weighs for discretionary denial.

F. *Fintiv* Factor 6: The Petition Does Not Present Compelling Evidence Of Un-patentability

Factor 6—the merits of the Petition—also counsels for denial because, as detailed below, the petition fails to show a “reasonable likelihood” that Petitioner would prevail on any challenged claim. See 35 U.S.C. § 314(a). In fact, the Board has already rejected the arguments presented in this petition when it rejected an earlier IPR petition pressing the same primary reference (obviousness in view of Tahan and Menahem) and arguments against the ’869 patent’s parent (the ’844 patent). See *’844 Denial* at 17-22. The Board refused to institute trial on the *’844 Trend Petition* because “Petitioner ha[d] not established a reasonable likelihood of proving that at least one challenged claim of the ’844 patent is unpatentable over the cited prior art.” (*’844 Trend Denial* at 19.) Though the Board found that Tahan suggests turning features on or off at the classification stage (*id.* at 15-16), it found,

in relevant part, that Tahan does not (1) do so based on the values of static data points (*id.* at 17) or (2) “evaluat[e] the feature vector using support vector processing” (*id.* at 18). Because the petition fails to show a reasonable likelihood of success on the merits, the Petition may not be instituted.

Even if the Petition did have a “reasonable likelihood” of succeeding (it does not), it certainly would not pass the higher threshold of a “compelling” unpatentability challenge that could preclude discretionary denial. Guidance at 4. “[T]he compelling merits standard is a higher standard than the standard for institution set by statute.” *Commscope Techs. v. Dali Wireless, Inc.*, IPR 2022-01242, Paper 23, at 3 (PTAB Feb. 27, 2023). That higher standard requires that the Petition is “highly likely” to succeed on the merits. *Id.* at 4. The opposite is true here.

* * *

In sum, all six factors weigh for discretionary denial. Accordingly, Patent Owner respectfully requests that the Board exercise its discretion to deny institution.

III. THE PETITION IS NOT REASONABLY LIKELY TO SUCCEED.

The Board should deny institution because the petition fails to show a reasonable likelihood of prevailing on any challenged claim. The petition presents a single ground of obviousness against the independent claims (claims 1, 10, 15, 18,

32, and 46) and certain dependent claims, and it presents a second ground against certain dependent claims only:

Ground 1	Section 103(a)	Rendered obvious by the combination of Tahan in view of Menahem and Forman	Claims 1-4, 8, 10-25, 29, 32-39, 43, 46-53, and 57
Ground 2	Section 103(a)	Rendered obvious by the combination of Tahan, Menahem, and Schultz	Dependent Claims 5-7, 9, 26-28, 30-31, 40-42, 44-45, 54-56, and 58-59

Neither ground has a reasonable likelihood of succeeding.

Ground 1 fails on the independent claims:

First, the proposed combination fails to suggest a machine learning malware classifier wherein features of the input feature vector are selectively turned on or off at the classification stage. Though the Board has previously found that Tahan does teach that limitation, ('844 *Trend Denial* 15-16), that finding is based on an incorrect understanding of Tahan, and this Panel should decline to adopt it.

Second, the proposed combination fails to suggest that the turning on or off of the features in the feature vector is done based on “whether a value of one of the plurality of static data points [extracted from the executable being classified] is within a predetermined range.” As the Board already found in denying Petitioner’s petition against the ’844 patent, Tahan “discloses dropping segments based on

entropy and ... [other] indices, not the values of the segments themselves.” (’844 *Trend Denial* 17.)

Third, as the Board’s previous findings confirm (’844 *Trend Denial* 18-19), the proposed combination fails to suggest (1) “providing the generated feature vector to one or more support vector machines to build a probabilistic model,” as recited in independent claims 1, 10, and 15; or (2) “evaluating the feature vector using a machine learning model,” as recited in independent claims 18, 32, and 46.

Because Ground 1 fails for all independent claims (1, 10, 15, 18, 32), it also fails for all dependent claims.

Petitioner does not argue that Ground 2 remedies any of these shortcomings (Petition 58-59), so that ground fails as well.⁵

⁵ Because this is a preliminary response, Open Text does not address every limitation of every claim. But this is not an admission that any limitation not discussed is disclosed or rendered obvious by the cited references. Open Text reserves the right to argue that they are not should the Board institute trial.

A. Background

Every day, thousands of new executable files (“executables”) are created and distributed across computer networks. (EX-1001 1:18-19.) It is the job of computer security software to quickly recognize which of those executables is malicious or otherwise unwanted, and to prevent execution of them. (*Id.* 1:18-24.)

One conventional approach was to perform “static analysis,” whereby the security software analyzes the executable’s code to identify known malicious code blocks within the executable. (*Id.* 2:23-24.) That conventional approach had two drawbacks: (1) it could not identify malicious files that it had not seen before, and (2) it was slow at evaluating encrypted files because it required the security software to first decrypt and disassemble the code.

One solution to the first problem, *i.e.*, identifying malicious code that has not been seen before, is to use a “machine learning classifier.” (*Id.* 1:27-48.) Machine learning classifiers are software programs that are “trained” to classify new, unknown executables by studying a large group of *known* executables, *e.g.*, “known malicious executable files, known benign executable files, and [known] potentially unwanted executable files.” (*Id.*) During a “training stage,” the system studies known executables to identify features or combinations of features that are correlated with each category of executable, *e.g.*, malicious, benign, unwanted, etc. Based

on these identified features, the system builds a “classifier”—*i.e.*, a predictive model that accepts a set of values for the features as input and outputs a predicted classification. The model can then be used to classify previously unseen executables: the computer security software simply extracts from the previously unseen executable values for the features used by the classifier, inputs those values for the features into the classifier, and thus obtains a predicted classification for the executable. These are the steps of the “classification stage.” (EX- 2001 ¶¶ 22-28.)

A classifier is typically trained by studying a set of training samples (known executables) to identify a set of features that are useful for classifying future executables. From each executable, the system extracts values for the identified features and stores them in a “feature vector” representing the executable. The system then finds a mathematical function that best separates the feature vectors into respective categories. To classify a new sample, the system creates a feature vector representing the new sample—*i.e.*, the system analyzes the new sample to extract values for the preidentified set of features (a process known as “feature extraction”) and uses those values to create a feature vector representing the new sample. The system then inputs the feature vector into the classifier for classification. (EX- 2001 ¶¶ 23-28.)

Because the computational complexity (and therefore the required computational time and resources) grows with the number of features a classifier has been

trained to consider, the classifier training phase may sometimes include a process known as “feature selection” or “feature pruning.” Feature selection is the process of identifying the features that are most effective and/or computationally inexpensive, and configuring the classifier to consider only those features. Feature selection at the training stage reduces the computational burden at the classification stage because the classifier will have to extract fewer features and/or perform less complex mathematical computations. (EX-2001 ¶24.)

1. The '869 Patent

The '869 patent improves upon the prior art by introducing an executable classification technique that performs feature selection at the *classification stage, based on the values of static data points extracted from the executable being classified*, rather than only at the training stage. Performing feature selection during the classification stage allows the system to dynamically streamline its performance based on the executable being classified rather than always determining values for, and using, all features identified at the training stage.

The invention achieves feature selection during the classification stage by turning features “on” or “off” during that stage based on the content of the executable being evaluated. (EX-1001 7:43-8:7; 12:61-13:11.) For example, the classifier may consider certain features (at the classification stage) only if the size of the executable

being evaluated is above a given threshold or only if the executable includes a certain type of content, *e.g.*, copyright information. (*Id.* 7:43-8:7.)

To decide (at the classification stage) which features should be turned on or off, the system analyzes the values of “static data points” extracted from the executable. The specification teaches that a “static data point” may be a “point of reference,” such as the executable’s file size, section names, legal information, and so forth. (*See* EX-1001 5:61-6:44, 8:39-43.)

The specification provides many examples of static data points (*see id.* Table 1.1, 8:39-43), and a person of ordinary skill would recognize that the provided examples may correspond to fields defined by the “Portable Executable” (or “PE”) file format. (EX-2001 ¶¶29-30; EX-1001 3:43-54 (discussing the extraction of static data points from PE files as the preferred but not exclusive embodiment), *id.* 4:29-31 (describing extraction of static data points from PE files).) In such embodiments, a static data point may include a field name, *e.g.*, a predefined character string such as “File Size,” and a field value, *e.g.*, a number such as 4982784. (EX-1001 8:39.)

The claimed system can turn features on or off *during the classification stage* based on the values of the static data points extracted from the executable being classified. (*Id.* 5:61-6:44, 8:39-43, 7:43-8:7 (describing selectively turning on or off features during “evaluation”).) For example, if the value of the “File Size” static data

point is greater than a certain numerical threshold, then the classifier may determine values for fewer or additional features. (*Id.*)

The extracted static data points are ultimately used create one or more feature vectors that are input into a trained classifier. (*Id.* 8:43-9:26.) The claims recite two types of classifiers for ultimately analyzing the feature vector representing the “file to be evaluated.” (*Id.* 16:18.) The final step of independent claims 1, 10, and 15 is “providing the generated feature vector to one or more support vector machines,” whereas the final step of independent claims 18, 32, and 46 is “evaluating the feature vector using a machine learning model.”

Turning a feature “off” at the classification stage improves the classifier’s performance because it allows the classifier to not use that feature in its prediction model, and thus, the classifier need not waste time and resources determining a value for that feature and considering that value in the performance of its classification algorithm. (EX-2001 ¶25.) This approach contrasts with traditional machine learning approaches, where at the classification stage, the classifier had to determine and consider values for every feature predetermined at the training stage. (*Id.*)

Claim 1 recites the invented technique, including both the training and classification stages:

1. A computer-implemented method comprising:

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

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

identifying an executable file to be evaluated;

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

generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data,

the training data comprising files known to fit into one of the plurality of categories of files,

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

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

(EX-1001 (claim 1).)

The claimed method begins with a *training* stage in which a “knowledge module” identifies static data points and associates them with categories of executables,

e.g., “harmful” or “benign.” The method then continues to a classification stage, which calls for “identifying an executable file to be evaluated,” extracting “static data points from the identified executable,” and “generating a feature vector from ... [those] static data points.” The feature vector is generated using a classifier that was previously “trained”—*i.e.*, at the training stage—to classify the static data points based on training data comprising files known to fit into one of the ... categories of files.” One or more features of the feature vector are then “turned on or off” based on whether a value of at least one of the static data points extracted from the executable being evaluated is within a predetermined range. For example, the classifier might turn certain features “off” if the “File Size” of the executable is smaller than some predefined threshold. Finally, the classifier classifies the executable by using support vector processing on the resulting feature vector.

Independent claims 1, 10, 15, 18, 32, and 46 all recite systems and methods that turn features of a feature vector on or off at the classification stage based on the value of static data points extracted from the executable being evaluated. As mentioned above, claims 1, 10 and 15 require “providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files.” In contrast, claims 18, 32, and 46 require “evaluating the feature vector using a machine learning model

to determine whether the executable fits into one of the categories of files.”

2. The Board's Rejection of Petitioner's IPR Petition Against the Parent '844 Patent Based on Overlapping Art

On March 3, 2023, Petitioner petitioned for IPR of the parent '844 patent, raising a substantially similar prior art argument (obviousness in view of Tahan and Menahem) as Petitioner presents here in Ground 1 (obviousness in view of Tahan, Menahem, and Forman). *See Trend Micro, Inc. v. Webroot Inc.*, IPR2023-00662, Paper 1 (PTAB Mar. 3, 2023) (“’844 Trend Petition”).

On October 13, 2023, the Board refused to institute trial on the ‘844 *Trend Petition* because “Petitioner ha[d] not established a reasonable likelihood of proving that at least one challenged claim of the ‘844 patent is unpatentable over the cited prior art.” (‘844 *Trend Denial* at 19.) Though the Board found that Tahan suggests turning features on or off at the classification stage (*id.* at 15-16), it found, in relevant part, that Tahan does not (1) do so based on the values of static data points (*id.* at 17) or (2) “evaluat[e] the feature vector using support vector processing” (*id.* at 18).

The '869 patent is a continuation of the '844 patent. Like the '844 patent, the '869 patent claims (1) feature selection at the classification stage based on the values of static data point, and (2) evaluation of that feature using a “support vector machine” or another “machine learning model.”

'844 (Cl. 1)	'869 Patent (Cl. 1)	'869 Patent (Cl. 18)
extracting a plurality of static data points from an executable file	extracting, by the knowledge module, a plurality of static data points from the identified executable file	extracting a plurality of static data points from the identified executable file;
generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points	generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points	generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points
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	wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range	wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range
evaluating the feature vector using support vector processing to determine whether the executable file is harmful	providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files	evaluating the feature vector using a machine learning model to determine whether the executable file fits into one of the categories of files

3. The Board's Rejection of Petitioner's Co-Defendant's PGR Petition Against the '869 Patent.

On November 29, 2023, the Board rejected a PGR petition filed by Petitioner's co-defendant (Sophos) against the '869 patent. (*Sophos Ltd. et al. v. Open*

Text, PGR-2023-0031, Paper 7 (PTAB Nov. 29, 2023) (“’869 *Sophos Denial*”).)

Though that petition did not involve the same prior art references asserted in this Petition, the denial includes at least two findings relevant here:

- The Board held “that the claims [of the ’869 patent] are directed to the classification stage.” (’869 *Sophos Denial* 22-23.)
- The Board held that a feature that is “turned ... off,” as recited in the independent claims, “is not used in the [classifier’s] evaluation.” (*Id.* at 25.) Specifically, Sophos had argued that a particular reference (Mori) teaches turning a feature “off” by setting the feature’s value to 0, and the Board rejected that argument on the basis that “we see nothing in Mori to suggest that a feature in the vector assigned a value of 0 *is not used in the evaluation.*” (*Id.* (emphasis added))

Both findings are relevant to and buttress Patent Owner’s analysis of Ground 1 here.

4. The Board’s Rejection of Petitioner’s Co-Defendant’s IPR Petition Against the Parent ’844 Patent.

Finally, also relevant here is the Board’s August 14, 2023 refusal to institute trial on an IPR petition filed by Petitioner’s co-defendant Sophos against the parent ’844 patent. (*Sophos Ltd. et al. v. Webroot*, IPR2023-00528, Paper 14 (PTAB August 14, 2023) (“’844 *Sophos Denial*”).) Though that petition did not involve the

same prior art asserted in this Petition, the Board's denial is relevant because it too includes an express finding "that the claims [of the '844 patent] are directed to the classification stage" and that the cited prior art failed to teach or suggest a "classification method in which one or more features of a feature vector generated from a plurality of static data points extracted from a file being analyzed are selectively turned on or off based on the value of the static data point." (*Id.* at 18.)

5. Claim Construction

The parties agree that no claim terms need be construed for this petition, and that all claim terms should carry their therefore plain and ordinary meaning in light of the intrinsic record. (Petition at 9-10.)

As referenced above, the parties are involved in parallel litigation where the district court has issued an interlocutory Claim Construction Order rejecting various proposals advanced by Petitioner and other defendants. The Board's regulations require that the Board consider those constructions. 37 C.F.R. § 42.100(b).

In analyzing the '869 patent, the district court construed two terms:

- **"the static data points"** The district court held that "static data points" is not indefinite and that the phrase takes its plain and ordinary meaning. (EX-2009 19.)

- **“within a predetermined range”** The district court held that the phrase “within a predetermined range” carries its plain and ordinary meaning, and it defined “range” as a “numerical range or a subset of a ‘particular scale’ with at least a lower limit or an upper limit, but not necessarily both.” (EX-2009 19.)

In analyzing the parent ’844 patent, the district court construed four additional terms that are relevant here because they appear also in the ’869 patent:

- **“character strings” being “within a predetermined range”** The district court held that it was not indefinite for a “character string” to be “within a predetermined range,” as recited in claim 17, and that this phrase should be given its plain and ordinary meaning. (EX-2007 18.)
- **“executable file”** The district court held that the phrase “executable file” carries its plain and ordinary meaning. (*Id.* 17.) The court rejected a proposal that would have required “executable files” to be “encrypted or packed.” (*Id.*)
- **“support vector processing”** The district court held that “support vector processing” carries its plain and ordinary meaning, and that this meaning is “processing that involves a support vector.” (*Id.* 18) The court rejected

arguments that “support vector processing” is limited to “processing with a support vector machine.” (*Id.*)

- **“at least one processor”** The district court held that this phrase has its plain and ordinary meaning, and it rejected the argument that the phrase is a means-plus-function limitation under § 112(f). (*Id.* 19.)

6. The Level of Ordinary Skill

For purposes of this proceeding, Patent Owner does not dispute Petitioner's articulation of the level of ordinary skill. (Petition 9.)

B. Overview of the cited art.

1. Overview of Tahan

Tahan is an academic paper that discusses various machine learning approaches for detecting malware in executables. (EX-1005 at 1.) After an “Introduction” in which Tahan describes background concepts in machine learning (including feature selection), Tahan proposes a malware classification system called “Mal-ID.” (EX-1005 at 5-13.)

Tahan teaches that the “basic insight” that motivated Mal-ID is that “both legitimate developers and hackers” use the same “higher level development platforms” and “the same specialized libraries.” (EX-1005 at 5-6.) Consequently, the “main idea” of Mal-ID is “to detect the file segments that originated from the development

platform or from a benign third-party library (termed here the Common Function) and then disregard those segments” while “the remaining segments would be compared to determine their degree of resemblance to a collection of known malware.” (EX-1005 at 7.) Tahan calls this process “common segment analysis.” (*Id.*)

The first variation of Mal-ID that Tahan discloses is called “Mal-ID basic.” (EX-1005 at 9-11.)

Mal-ID basic proceeds in two phases: (1) a “setup phase,” in which data is gathered from a training set of executables, and (2) a “detection phase,” in which “a previously unseen” executable is analyzed and classified using the data gathered in the setup phase. (EX-1005 at 7.)

Setup Phase

In the setup phase, a training set of known executables is analyzed to determine (1) the Common Functions (*i.e.*, those found in benign executables) and (2) the “Threat Functions” (*i.e.*, those found in malicious executables). (*Id.*)

The system starts by “collecting two kinds of files: benign and malware.” (EX-1005 at 7.) Then, a “CFL repository [containing the Common Function segments] is constructed from benign files and [a] TFL repository [containing the Threat Function segments] is constructed from malware files.” (*Id.*) “The CFL is constructed first and then the TFL.” (EX-1005 at 9.)

To construct the CFL, each benign “file is divided into 64 equal length segments” (*id.*), which are each further “broken into 3-grams (three consecutive bytes)” (EX-1005 at 7, 9.) Moreover, “[t]he training files are randomly divided into 64 groups.” (*Id.* at 7.)

Then, for every combination of 3-gram and file group, the system records in the CFL (1) whether the 3-gram appears in the files of that group (“3-gram-files-association”) and (2) which segment of the file the 3 gram appears in (“3-gram-relative-position-within-file”). (EX-1005 at 8-9.)

The TFL is then constructed: “Each file from the malware collection is broken into ... 64-byte segments,” “[e]ach segment is broken into 3-grams and then tested against the CFL” such that “[s]egments that are not in the CFL are added to the TFL.” (EX-1005 at 9.) “[T]he end result is the TFL, a repository made of segments found only in malware and not in benign files.” (*Id.*)

Detection Phase

In the detection phase, the data collected in the setup phase is analyzed to classify a previously unseen executable. (EX-1005 at 9-10.)

At a high level, the classification algorithm simply (1) “[d]ivide[s] file F into S segments of length L” and (2) determines whether any of those segments meets a given set of criteria. (EX-1005 at 10.) That set of criteria is:

- Unique Segment. “[A]ny duplicated segments are removed.” (*Id.*)
- Entropy Is Within Range. “[I]f the entropy is smaller than EntropyLow threshold or entropy is larger than Entropy-High ... then discard the segment.” (*Id.*) “Entropy” refers to “the entropy measure of the bytes within a specific segment candidate.” (EX-1005 at 9.)
- Not in CFL. “[I]f the CLF-MFG [(i.e., maximum total number of file-groups in CFL that contain the segment)] is larger than zero, then discard the segment.” (EX-1005 at 10.)
- In TFL. “[I]f the TFL-MFG index [(i.e., maximum total number of file groups in TFL that contain the segment)] is smaller or equal to the ThreatThreshold ... then discard the segment.” (*Id.*) For example, “only segments that appear two times or more [in the TFL] remain.” (*Id.*)
- TFL-Spread Within Range. “If the TFL-Spread index equals zero or if it is larger than what we term SR threshold, then discard the segment.” (*Id.*) The TFL “spread” “represents the spread of the signature’s 3-grams along the various areas for all the files in [the TFL].” (EX-1005 at 9.)

- Second Level Index Aggregation. The segment must be “found in malware and not in the CFL.” (EX-1005 at 11.)

To decide whether the executable is malicious or benign, the system simply counts how many of the executable's segments meet the above set of criteria. (EX-1005 at 11.) “If there are at least X segments” that meet the above set of criteria, the system determines that “the file is malware,” and “otherwise, [it] consider[s] the file as benign.” (*Id.*) Tahan discloses that “X” is chosen arbitrarily (e.g., by the system administrator). (*Id.*) For purposes of the experiments described in Tahan, the authors “implemented Mal-Id with X set to 1.” (*Id.*)

Tahan describes the executable's 3-grams as the extracted “features,” and the “entropy,” “MFG,” and “Spread” measures calculated based on those 3-grams as “meta-features.” (EX-1005 at 9 (“Note that the features, as described above, are in fact *meta-features* as they are used to represent *features of features* (features of the basic 3-grams).”) (emphasis in original).)

Importantly, Mal-ID Basic does not turn any features “off” *either* at the “setup phase” or at the “detection phase.” (EX-2001 ¶65.) At the “setup phase,” Mal-ID Basic identifies the “Common Function segments” and “Threat Function segments,” but it does not *prune* those sets of segments to improve subsequent classification performance, so it does not turn any features “off” at the “setup phase.” (EX-1005

at 7.) And during the subsequent “detection phase,” Mal-ID Basic segments the executable into fixed-length chunks and evaluates *all* those chunks against *all* the “Common Function segments” and “Threat Function segments” identified during the “setup phase.” Thus, Mal-ID Basic does not turn any features “off” during the “detection stage” either.

Petitioner argues that determining whether an executable includes a segment that meets certain criteria (as is done in Tahan’s “detection stage”) means turning “off” segments that fail to meet those criteria (by “discarding” them) and turning “on” segments that meet the criteria (by not “discarding” them). (Petition 23.) This is wrong because the Tahan performs no *additional analysis* of only the non-discarded features. It simply determines whether any non-discarded features exist—if so, the executable is malware, and if not, the executable is not malware. Thus, the “discarded” and non-“discarded” segments undergo *the same evaluation*. Where the “detection stage” requires performing the same evaluation of every segment, there is no basis to consider any segment “selectively turned on or off,” as claimed.

2. Overview of Menahem

Menahem discloses ideas for combining multiple malware classifiers into an “ensemble.” (EX-1006 at 1.)

Like Tahan, Menahem starts with an “Introduction” describing the general

process of “detecting malicious executables using Machine Learning (ML) techniques.” (*Id.*) As described above, that process involves (1) a training stage in which “[m]alicious and benign files are represented by a vector of features” that “are used to train a classifier,” and (2) a classification stage (which Menahem calls the “detection phase”) in which “an unknown file can be classified” using the trained classifier. (*Id.*) Menahem teaches that some classifiers can use features “extracted from the PE header” of executables. (*Id.*)

Menahem observes that, because different classifiers are built using different “inducer algorithms” and different “types of features,” “each classifier will perform differently,” such that some classifiers may be better or worse at classifying certain types of malware. (EX-1006 at 2.)

Menahem suggests that this problem can be solved by combining multiple types of classifiers into “multi-inducer ensembles.” (*Id.*) “The main idea of an ensemble methodology is to combine a set of classifiers in order to obtain a better composite global classifier.” (*Id.*) For example, Menahem discloses a “Majority Voting” schema in which a group of different classifiers “vote” and the classification chosen is “the class that obtains the highest number of votes.” (EX-1006 3.) Menahem also proposes more complex ways of combining the classifiers, including techniques called “stacking” and “Troika.” (EX-1006 4; EX-2001 ¶75.)

Menahem teaches that, “to make the ensemble more effective, there should be some sort of diversity between the classifiers,” and that this “diversity is obtained by using different types of inducers.” (EX-1004 at 3.)

3. Overview of Forman

Forman discloses a general-purpose machine-learning classification technique that uses what Forman calls “feature scaling.” (EX-1018 1:1-3.)

As shown in Figure 1 below, Forman’s technique follows the usual machine learning pattern of a “training” stage (elements 12, 14, and 16) followed by a “classification” stage (elements 22, 24, 18, and 27).

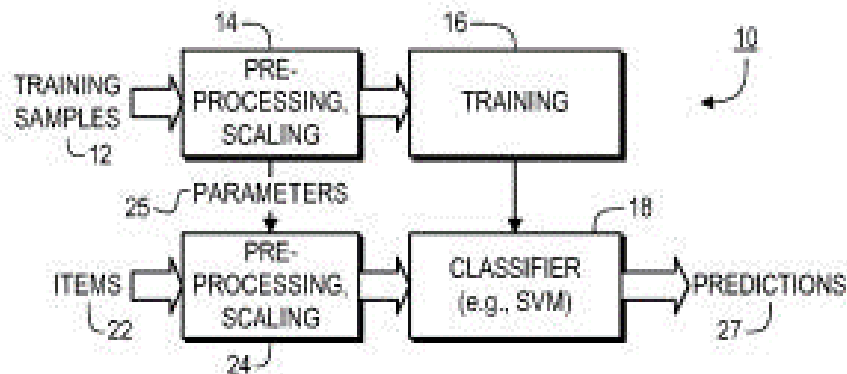


FIG. 1

(*id.* at Fig. 1.)

The patented system starts with a training stage. (EX-1018 2:27-3:55 and 6:49-7:7.) The training stage begins by extracting feature vectors from the received

“training samples” and then “pre-processing” those feature vectors. (*Id.* at 3:11-42.) The “preprocessing” can include traditional “feature selection” whereby features that are not sufficiently “predictive”—*i.e.*, have a low predictiveness “score”—are pruned. (*Id.* at 3:11-42.) Forman adds that the “preprocessing” can also include “scaling,” whereby the value of each feature is modified based, in part, the feature’s predictiveness score. (*Id.* at 7:8-38; *see also* EX-2001 ¶¶79, 81.) The preprocessed (*i.e.*, pruned and scaled) feature vectors are then used to train a classifier. (*Id.* at 3:67-4:3.)

Forman’s system then performs a classification stage. (EX-1018 3:56-4:3, 7:56-8:31.) In that stage, the system extracts a feature vector from the executable being classified and performs the same feature-selection and scaling that was performed during the training stage. (*Id.*) Importantly, the classification-stage feature-selection simply selects the same features that were selected at the training stage. (EX-2001 ¶80.) It does not prune features based on the values of the features extracted from the executable being classified. (*Id.* ¶81.) Foreman’s system then scales the values of the remaining features, as was done at the training stage. (*Id.*) The preprocessed feature vector for the executable being classified is then fed into the classifier for classification. (EX-1001 8:43-9:54.)

C. Legal Standards

“In an inter partes review, the burden of persuasion is on the petitioner.... Failure to prove the matter as required by the applicable standard means that the party with the burden of persuasion loses on that point—thus, if the fact trier of the issue is left uncertain, the party with the burden loses.” *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378-79 (Fed. Cir. 2015); *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (noting petitioners have the “burden from the onset to show with particularity why the patent it challenges is unpatentable”). Accordingly, a petition must include “a detailed explanation of the significance of the evidence.” 37 C.F.R. § 42.22(a)(2); *see also* 35 U.S.C. § 312(a)(3); *Lumenis Be Ltd. v. BTL Healthcare Techs. A.S.*, IPR2021-01280, Paper 8 at 22 (PTAB Mar. 9, 2022) (finding petitioner had not met its burden and declining to institute because petitioner’s argument lacked explanation and the Board was unwilling to speculate).

D. Ground 1: The combination of Tahan, Menahem, and Forman did not render claims 1-4, 8, 10-25, 29, 32-39, 43, 46-53, and 57 obvious.

Petitioner fails to establish a reasonable likelihood that the combination of Tahan, Menahem, and Forman rendered claims 1-4, 8, 10-25, 29, 32-39, 43, 46-53, and 57 obvious because the combination fails to teach or suggest: (1) turning features

of a feature vector on or off at the classification stage, (2) doing so based on the values of static data points extracted from the executable being within a predetermined range, or (3) evaluating such a feature vector using a “support vector machine” (as recited in claims 1, 10, and 15) or a “machine learning model” (as recited in claims 18, 32, and 46). (EX-2001, ¶84.)

As to the first limitation identified above, Patent Owner respectfully requests that the Board reconsider its finding in the *'844 Trend Denial* that Tahan teaches turning features on or off at the classification stage. (*'844 Trend Denial* 15-16.) Tahan does not teach that limitation, and combination with Menahem and Forman does not remedy this shortcoming.

As to the second and third limitations identified above, Patent Owner respectfully requests that the Board reaffirm its findings in the *'844 Trend Denial* that the combination of Tahan and Foreman fails to teach either turning features on or off based on the values of static data points extracted from the executable being within a predetermined range *or* evaluating such a feature vector using a “support vector machine” or other “machine learning model.” (*'844 Trend Denial* 17-19.) The combination of Tahan and Menahem does not teach either limitation, and the further combination with Forman does not remedy either shortcoming.

1. The proposed combination fails to teach turning features on or off at the classification stage.

None of the cited references, alone or in combination, teach the idea of turning features on or off at the classification stage.

i. Tahan

Petitioner is wrong that Tahan teaches turning features “on or off” at the classification stage (*i.e.*, Tahan’s “Detection Phase”). (Petition 23-24.) As the Board recognized in the *'869 Sophos Denial*, a feature that is “turned ... off” “is not used in the [classifier’s] evaluation.” (*'869 Sophos Denial* 25.) But Tahan’s “Detection Phase” calls for extracting and analyzing *every* segment (purported “features”) of the executable in the same way, meaning that *every* segment is used in the classifier’s evaluation. Tahan’s “Detection Phase” therefore does not turn “off” any features.

A benefit of turning features “on or off” at the classification stage is to optimize the classifier’s performance (*e.g.*, runtime and/or accuracy) based on the content of the executable being classified. (EX-2001 ¶87.) Using this approach, a classifier is not locked into extracting and evaluating the full set of features that was predetermined at the training stage, without regard to the content of the executable being classified at the classification stage. (EX-2001 ¶¶31-32.) Instead, the classifier can turn some features “off” and thus avoid the work and potential inaccuracies

introduced by extracting and considering values for those features.

Tahan's system lacks these benefits because it performs *the same* classification-stage analysis regardless of the content of the executable being classified. Specifically, Tahan's classification stage ("Detection Phase") extracts *every* segment in the executable and tests *every one* to determine if any segment satisfies a given set of criteria. (EX-1005 at 10-11.). If at least one segment meets the criteria, the executable is classified as malware. (*Id.*) If not, the executable is classified as benign. (*Id.*) Because every segment is extracted and considered in the same way, no segment is "turned on or off." (EX-2001 ¶88.) Rather, *all* segments are "on." (*Id.*)

Petitioner's confusion stems from Tahan's *labeling* of segments that fail to meet the criteria as "discarded." (Petition 23; '844 *Trend Denial* at 15-16.) As the Board mistakenly did in the '844 *Trend Denial*, Petitioner equates labeling a segment "discarded" with turning it "off." But that is wrong because segments labeled "discarded" are considered by the classifier no less than segments that are not.

Labeling a segment "discarded" does not exclude the segment from some further consideration that segments not labelled "discarded" undergo. (EX-2001 ¶ 90) As noted above, *every* segment undergoes the *same* processing, which results in *receiving* the label "discarded" or not. (*Id.*) After all segments have been labeled, the classifier merely asks whether *all* segments received the "discarded" label. (*Id.*) The

classification decision thus turns equally on *all* segments. (*Id.*) The segments labeled “discarded” are therefore not “turned ... off” in any sense. (*Id.*) Tahan therefore does not teach turning features off at the classification stage.

ii. Combination with Menahem

The addition of Menahem does not remedy the shortcomings discussed above, and Petitioner makes no effort to argue otherwise. (Petition 24.) The petition devotes only one passing sentence to explain why combining Menahem with Tahan might suggest turning features on or off at the classification stage: after noting that Tahan’s decisions to apply the “discarded” label to certain segments may involve considering the segment’s “gain ratio,” Petitioner notes that “Menahem describes similar compatible selection functionality.” (Petition 24.) In support, Petitioner cites the expert declaration of Dr. Nielson, who opines that “Menahem also discloses feature selection *during training*,” including by considering “gain ratio.” (EX-1003 ¶¶277-82 (emphasis added).) Thus, Petitioner does not even argue, let alone show, that Menahem teaches feature selection at the classification stage or that combining Menahem with Tahan rendered that functionality obvious.

The portion of Menahem Dr. Neilson cites—Section 3 (EX-1006 at 6-7)—confirms his opinion that Menahem “discloses feature selection *during training*.” (EX-1003 ¶¶277-82 (emphasis added).) As the outline of Menahem’s Section 3

explains, Sections 3.2 and 3.3 of Menahem “describe the *training process* and the datasets we had used.” (EX-1006 at 5 (emphasis added).) Section 3.2 discloses “trimming the number of features” by “identify[ing] those terms that appear most frequently in files” *of the training set*. (EX-1006 at 5.) Indeed, the disclosure of Section 3.2 *precedes* the description of “[t]he training process” in Section 3.3, which details “the training of the base-classifiers.” (EX-1006 at 6.) Thus, Section 3 of Menahem does not describe turning features on or off at the classification stage. (EX-2001 ¶92.)

In sum, Petitioner does not, and cannot, argue that adding Menahem to Tahan would have rendered it obvious to turn features on or off at the classification stage.

iii. Combination with Forman

New in this petition is an attempt to augment Tahan with Forman’s teaching of “pre-processing” feature vectors at the classification stage. (Petition 24-28.) Petitioner argues that Forman’s classification-stage “pre-processing” can include feature selection, and that it would therefore have been obvious to modify Tahan to “discard” segments “also during classification.” (*Id.* at 25-26; EX-2001 ¶¶94-95.)

This argument fails because it purports to remedy the wrong problem: the trouble with Tahan is not that the “discarded” labeling occurs outside the classification stage, but that the labeling (which *does* occur at the classification stage) does

not turn features on or off. In any event, Petitioner is wrong that Forman teaches feature selection at the classification stage. (EX-2001 ¶95.)

First, Petitioner's proposed combination addresses the wrong problem with Tahan. Petitioner's argument is that Forman would have made it obvious that Tahan's labeling of a segment as "discarded" can occur at the classification stage. (Petition 25.) But the problem with Tahan is not that segments are labelled "discarded" *outside* the classification stage. Indeed, the labels *are* applied at the classification stage. Instead, the problem is, as discussed above, that the labeling does not turn the segment "off," within the meaning of the claims. The Board should therefore reject the proposed combination with Forman on the basis that it fails to suggest how Tahan might be modified to turn features off at the classification stage. *See Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (The Petitioner has the "burden from the onset to show with particularity why the patent it challenges is unpatentable."); *Everstar Merch. Co. v. Willis Elec. Co.*, No. 2021-1882, 2022 WL 1089909, at *3 (Fed. Cir. Apr. 12, 2022) ("A petitioner may not assert an 'entirely new rationale' in a post-institution reply").

Second, Petitioner is anyway wrong that Forman alone teaches turning features on or off at the classification stage. (Petition 25-26.) Instead, just as in traditional systems, Forman identifies the "feature set" to be used *at the training stage*,

based on which features are most “predictive” at that stage, and it includes no disclosure of a separate, independent feature selection process (or criteria) for the classification stage. (EX-1018 2:59-3:42; *see also id.* at 3:11-14, 6:52-7:7.)

In Forman’s training stage, “training samples 12, initially are input into a pre-processing section 14,” and “[e]ach of the training samples preferably is represented by values for features in a designated feature set.” (*Id.* at 2:29-33; *see also id.* at 6:53-55.) The training stage then executes a “pre-processing” step 14 that performs feature selection, for example determining and pruning the least “predictive” features from the original designated feature set. (*Id.* at 2:27-3:42.) Finally, the “*modified feature sets 38* are input into training section 16” to train the classifier. (*Id.* at 3:44-45.)

In Forman’s classification stage, the classifier uses *the same* “feature set 38” identified in the training stage. Indeed, the item to be classified (“item 22”) “is represented by a set of values corresponding to a *feature set 38*”—*i.e.*, the same feature set identified in the training stage. (*Id.* at 7:62-63 (emphasis added).)

Though Forman does illustrate the classification stage as having its own “pre-processing” step 24, the role of that “pre-processing” step 24 is simply to extract the same feature set that was determined at the training stage. Indeed, Forman discloses that the “processing parameters” “generated in [training-stage] pre-processing

section 14 [are] provided for use by pre-processing section 24,” which “performs processing that is *identical or at least substantially similar* to that of [training-stage] pre-processing section 14.” (*Id.* at 3:56-60.) Thus, classification-stage “pre-processing” step 24 is simply a traditional feature *extraction* step—that is, the process of extracting values for the same “feature set 38” identified at the training stage. The choice of *which* features to select (i.e., feature selection) is not made at the classification stage based on anything seen at that stage, but rather, at the training stage, based on feature effectiveness, as is traditional. (EX-2001 ¶101.)

To whatever extent Forman's classification-stage “pre-processing” can be characterized as turning features on or off (it cannot), the Board should still reject Ground 1 on the basis that (1) as discussed above, Petitioner has failed to articulate how the combination of Tahan and Forman would turn features on or off at the classification stage, and (2) as discussed below, neither Forman's nor Tahan's purported turning of features “on or off” is based on the values of static data points extracted from the executable being classified.

2. The proposed combination fails to teach turning features on or off based on the values of static data points extracted from the executable being classified.

The Board should deny institution based on a second, independent ground: as the Board found in the '844 *Trend Denial*, Tahan's decision to label certain segments

as “discarded” is not based on *the value of* static data points extracted from the executable being classified, but rather, on the values of certain “meta-features” of the segments. (*'844 Trend Denial* 17.) Neither Menahem nor Forman teach or suggest that the selection be based on the value of static data points extracted from the executable being classified. Indeed, Petitioner does not even argue that either of those secondary references suggest that feature. Thus, the proposed combination fails to teach or suggest this limitation.

i. Tahan

Regarding Tahan, Petitioner argues that “Tahan teaches selecting segments (features / static data points) of files for the feature vector *based on an entropy or CFL-MFG index value of the segment.*” (Petition 23 (emphasis added).) But as the Board found in the *'844 Trend Denial*, a decision “based on entropy and the CFL-MFG, TFL-MFG, and TFL-Spread indices” is not a decision based on “the values of the segments themselves.” (*'844 Trend Denial* 17.) As the Board reasoned there, “Petitioner does not explain how or why the meta-features or the indices [in Tahan] should be considered ‘a value of one or more static data points,’” as claimed. (*Id.*) The Board should reach the same conclusion here.

Further regarding Tahan, Petitioner also argues that Tahan’s decision to apply the “discarded” label to a segment may turn on the “gain ratio” of the segment

(Petition 24.) According to Petitioner, “gain ratio” is a value reflecting the degree to which “the additional segment adds any signal gain in helping [the classifier] identify malware or whether the segment is [instead] noise.” (*Id.*)

By Petitioner's own description then, a gain ratio is a measure of effectiveness, not the value of the segment. Consequently, Tahan's use of a “gain ratio” does not show that Tahan turns any features “on or off” *based on the value of a static data point extracted from the executable being classified*. And indeed, the Board reached exactly that conclusion when it rejected Petitioner's identical argument in the '844 *Trend Denial*. (*Trend Micro Inc.*, IPR2023-00662, Paper 1 at 21 (PTAB Mar. 3, 2023) (Petitioner arguing that Tahan's use of “gain ratio” teaches turning features on or off based on the values of static data points).

ii. Combination with Menahem

The addition of Menahem does not show that it would have been obvious to modify Tahan to turn features on or off *based on the values of static data points extracted from the executable being classified*. And again, Petitioner does not even make that argument. (Petition 24.)

As discussed above, Petitioner's only use of Menahem against the “turned on or off” limitation is to note that Menahem's training-stage feature selection may consider a feature's “gain ratio.” (Petition 24.) And as shown in the previous section,

Petitioner describes “gain ratio” as a measure of a feature’s classification *effectiveness*, not the value of the feature. Thus, Petitioner does not argue that Menahem teaches feature selection *based on the values of static data points extracted from the executable being classified*.

Though the Petition does not discuss it, Petitioner’s expert also cites Menahem’s use of “Inverse Document Frequency” (“TF-IDF”) as a value of training-stage feature selection. (EX1003 ¶¶279, 280.) But, as Petitioner’s expert admits, “‘TF-IDF values’ represent values *derived from* n-grams,” not the value of the n-grams themselves. (*Id.* ¶280.) Thus, like Tahan’s entropy values, Menahem’s TF-IDF values are meta-features. As it held with respect to Tahan’s “meta-features,” the Board should hold that Menahem’s meta-features fail to satisfy the value limitation because “Petitioner does not explain how or why the meta-features ... should be considered ‘a value of one or more static data points.’” (*’844 Trend Denial* 17.)

iii. Combination with Forman

Finally, the addition of Forman also fails to teach feature selection *based on the value of static data points extracted from the executable being classified*. And once again, Petitioner does not argue otherwise. (Petition 24-25.)

Petitioner argues that Forman teaches classification-stage feature selection

that is “identical or at least substantially similar to that of [training-stage] pre-processing section 14.” (Petition 25 citing EX1003, ¶283 quoting EX-1018, 3:56-60.) But Forman teaches that the training-stage “pre-processing section 14” performs “feature selection” based on “scores” of how “*predictive*” the features are. (EX1018 2:59-3:42, 3:11-14, 6:52-7:7.) Those “scores,” which “generated in pre-processing section 14[, are] provided for use by pre-processing section 24.” (*Id.* at 3:60-65.) Thus, Forman’s classification-stage “pre-processing” step 24 selects features based on those same predictiveness scores. It does not do so based on the values of static data points extracted from the sample being classified at the classification stage.

In fact, Forman never suggests that the classification-stage “pre-processing section 24” selects features based on *anything* found in the executable being classified, let alone based on the values of static data points extracted from that executable. For example, Forman does not disclose any parameters used by classification-stage “pre-processing section 24” other than those received from training-stage “pre-processing section 14.”

Thus, even if Forman’s classification-stage “pre-processing” were considered turning features “on or off” (which it is not), Forman would still fail to teach doing that based on the values of static data points from the executable being classified. The addition of Forman to Tahan and Menahem therefore does not teach or suggest

turning features on or off *based on the values of static data points extracted from the executable being classified at the classification stage.*

3. The proposed combination fails to teach evaluation of the recited feature vectors using a “support vector machine” or any other “machine learning model.”

Finally, the Board should deny institution based on a third, independent ground: as the Board found in the *'844 Trend Denial*, even if Tahan's marking of certain segments as “discarded” constitutes turning those segments “off” (it does not), and even if Tahan were to make those marking decisions based on the values of static data points extracted from the executable being classified (it does not), Tahan would still not teach evaluating the non-discarded segments using a “support vector machine” (as recited in claims 1, 10, and 15) or a “machine learning model” (as recited in claims 18, 32, and 46). (*'844 Trend Denial* 18-19.) As the Board found in the *'844 Trend Denial*, Tahan “uses a different method to evaluate the feature vector”—i.e., simply counting the number of non-“discarded” segments. (*'844 Trend Denial* 19.) Thus, Tahan fails to teach the final limitation of each independent claim, which requires providing the modified feature vector to a “support vector machine” or otherwise “evaluating” it using some other “machine learning model.” The Board has already held that combination with Menahem fails to remedy these shortcomings (*id.* at 19), and it should hold the same for the proposed further combination

with Forman introduced here.

i. Tahan

In Tahan's Mal-ID system, the purported "feature vector" whose features were purportedly "turned on or off" (*i.e.*, labeled "discarded" or not) is not subsequently evaluated by an SVM or any other machine learning model, as recited. Instead, after the Mal-ID system decides which segments are "discarded" (*i.e.*, purported turning features on or off), it makes its classification decision based simply on whether any segment was *not* "discarded." (EX-1005 10-11.) Determining that at least one segment is not "discarded" does not involve any machine learning models, let alone SVMs. (EX-2001 ¶116.) Rather, it is a simple, hard-coded *if-then* statement based on no previous learning. (*Id.*)

Petitioner argues briefly that it would have been obvious for Mal-ID's classification decision to be made using an SVM or other machine learning model instead of the hard-coded *if-then* statement because "Tahan discloses that support vector machines (SVMs) are a known classification technique." (Petition 28.) But as the Board has already explained when it rejected Petitioner's identical argument in the '844 *Trend Denial*, "the fact that SVMs were used in existing systems is insufficient to show that it would have been obvious to use a SVM, or SVP, in Tahan's Mal-ID system, where Mal-ID uses a different method to evaluate the feature vector to

determine whether the executable file is harmful.” (*'844 Trend Denial* 19.)

The Petition includes nothing new on this issue with respect to Tahan. Consequently, the Board should again reject Petitioner's unsupported argument that the mere existence of SVMs and machine learning models in the prior art would have made it obvious to modify Mal-ID's classification decision to replace its hard-coded if-then statement with some type of machine learning model.

ii. Combination with Menahem

Adding Menahem fails to remedy the problems with Tahan. (Petition 28-29.) Petitioner relies on Menahem's disclosure of SVMs generally and of a “composite classifier,” wherein the output prediction from a “base classifier” becomes the input into a separate, “final classifier.” (Petition 29.) But Petitioner is entirely unclear about how these teachings might be combined to suggest that Tahan's purported reduced feature vector (which is not Tahan's final output) would become the input into an SVM or other machine learning model. The corresponding portions of Petitioner's expert declaration provide no additional clarity. (EX-1003 ¶¶296-300.)

Petitioner's argument here appears to be the same one it made in its earlier, failed petition against the parent '844 patent. (See *Trend Micro Inc. v. Webroot Inc.*, IPR2023-00662, Paper 1 at 51-53 (PTAB Mar. 3, 2023). And, as in those proceedings, the Board should reject the argument because “the cited portions of Menahem

are, as in Tahan, background describing the state of the art, and Petitioner fails to explain how those disclosures might be combined with a method like Mal-ID.” (’844 *Trend Denial* 19.)

To the extent Petitioner is suggesting that it would have been obvious to create a “composite classifier” where the output of Tahan’s Mal-ID system is the input to an SVM or other machine learning model, this argument fails because the output from Tahan’s Mal-ID system is a classification, not the purportedly pruned feature vector. (EX-2001 ¶121.) As Petitioner’s expert observes (EX-1003 ¶296), Menahem’s idea is to “ensemble the *results* of multiple classifiers,” for example, using a “Majority Voting Schema.” Petitioner fails to explain what in Menachem would have prompted a POSITA to replace Tahan’s decision step with a step that instead passes an *intermediate* output (i.e., Tahan’s purportedly pruned feature vector) into a separate classifier (such as an SVM), let alone *how* that would work or *why* a POSIT would have had a reasonable expectation of success.

The Board should reaffirm its previous holding that Petitioner has failed to explain how the combination of Menahem with Tahan would have taught that the purported feature vector in Tahan should be provided to an SVM or otherwise evaluated by any other machine learning model.

iii. Combination with Forman

Combination with Forman does not resolve any of the shortcomings of Tahan and Menahem described above. Petitioner's only argument regarding Forman on this limitation is a fleeting note that Forman evaluates feature vectors using an SVM classifier. (Petition 30.) But as the Board has already explained, "the fact that SVMs were used in existing systems is insufficient to show that it would have been obvious to use a SVM, or SVP, in Tahan's Mal-ID system, where Mal-ID uses a different method to evaluate the feature vector to determine whether the executable file is harmful." (*'844 Trend Denial* 19.) Thus, Petitioner's observation that Forman used SVM classifiers in the traditional way did not render it obvious to replace Tahan's evaluation method with an SVM, let alone explain how that replace would work.

For at least these reasons, the combination of Tahan, Menahem, and Forman does not suggest evaluating the claimed feature vector using a "support vector machine" (as recited in claims 1, 10, and 15) or a "machine learning model" (as recited in claims 18, 32, and 46). (EX-2001 ¶¶123-124.)

* * *

As shown above, neither Tahan nor Menahem nor Forman (alone or in combination) suggest an ML malware classifier with (1) the flexibility to turn features on or off at the classification stage, (2) do so based on the values of static data points

extracted from the executable being evaluated at that stage, and (3) provide the resulting feature vector to an SVM or other machine learning model. Accordingly, the proposed combination of these references does not render any of the challenged claims obvious.

Because Petitioner has failed to establish a reasonable likelihood of success on any of the independent claims, it cannot meet its burden for any of the challenged claims that depend upon those independent claims. *Twinstrand Biosciences, Inc. v. Guardant Health, Inc.*, IPR2022-01115, Paper 14, at 19 (PTAB Dec. 30, 2022). Accordingly, Petitioner has failed to show a reasonable likelihood of prevailing on Ground 1, with respect to any of the challenged claims.

E. Ground 2: The combination of Tahan, Menahem, and Schultz did not render dependent claims 5-7, 9, 26-28, 30-31, 40-42, 44-45, 54-56, and 58-59 obvious.

Ground 2 fails for similar reasons as Ground 1. The Ground adds Schultz only to purportedly teach the limitations *added* by dependent claims 6, 7, 9, 15, 16, 18, and 20. (Petition at 60-70.) Because Ground 2 includes no argument that Schultz remedies any of the shortcomings of Ground 1 identified above with respect to the independent claims, and the claims Ground 2 challenges all depend on those independent claims, Ground 2 cannot render any of the challenged claims obvious.

IV. THERE ARE OBJECTIVE INDICIA OF NON-OBVIOUSNESS

In determining obviousness of the claimed invention, objective evidence of secondary considerations pertaining to non-obviousness must be considered, including long-felt but unsolved needs, failure of others, and commercial success. *In re Huang*, 100 F.3d 135, 139 (Fed. Cir. 1996) (citing *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966)). These indicia constitute independent evidence of non-obviousness that can be the most probative evidence in that regard “in avoiding the trap of hindsight when reviewing, what otherwise seems like, a combination of known elements.” *Leo Pharm. Prods., Ltd. v. Rea*, 726 F.3d 1346, 1358 (Fed. Cir. 2013).

The Board has repeatedly cautioned “that known evidence of secondary considerations should be addressed in the petition.” *See, e.g., Robert Bosch Tool Corp. v. SD3, LLC*, IPR2016-01751, Paper 15, at 23 (PTAB Mar. 22, 2017) (collecting prior Board decisions that deny institution based, at least in part, on a petitioner’s failure to address known evidence of secondary considerations). Despite Patent Owner providing evidence regarding the long-felt, unmet need for the invention in its district court complaint against Petitioner (*see, e.g.,* EX-2011 at 24-27, discussing in the context of the parent ’844 patent), Petitioner fails to consider such evidence in its petition; thus, institution should be denied.

As discussed above, a long-felt and persistent need existed to improve

malware classification performance by dynamically calibrating feature vectors at the classification stage based on the content of the executable. (EX-2001 ¶128.) Prior art attempts were unable to improve classification performance by performing such dynamic calibration at the classification stage. Where others failed, the inventors of the '869 patent succeeded.

Patent Owner currently accuses industry participants, including Petitioner, of infringing the innovative technology claimed in the '869 patent, demonstrating the critical need and widespread industry reliance on these novel methods. Petitioner, for example, markets and provide Apex One and its EDR/XDR products, the components of which practice the systems and methods claimed in the '896 patent.

Accordingly, secondary objective indicia of non-obviousness confirm that Petitioner cannot show a reasonable likelihood of success as to any of the challenged independent claims.

V. CONCLUSION

For the foregoing reasons, the Board should decline to institute IPR of the '869 patent.

Date: December 21, 2023

Respectfully submitted,

BY: /s/ Brian Eutermoser
Brian Eutermoser
Registration No. 64,058

CERTIFICATION OF WORD COUNT

The undersigned hereby certifies that the portions of the above-captioned **PA-
TENT OWNER'S PRELIMINARY RESPONSE** specified in 37 C.F.R. § 42.24 includes 13,522 words and therefore complies with the 14,000-word limit set forth in 37 C.F.R. § 42.24. This word count was prepared using Microsoft Word 365 v.2202.

Date: December 21, 2023

Respectfully submitted,

BY: /s/ Brian Eutermoser
Brian Eutermoser
Registration No. 64,058

Certificate of Service

The undersigned hereby certifies that the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** and all supporting **EXHIBITS 2001-2014** were served by filing the documents through the Patent Trial and Appeal Case Tracking System (P-TACTS) and courtesy copies were also served via electronic mail, in their entireties on the following:

Robert M. Hansen
Matthew J. Anderson
Scott B. Amankwatia
The Marbury Law Group, PLLC
11800 Sunrise Valley Drive, 15th Floor
Reston, VA 20191
rhansen@marburylaw.com
manderson@marburylaw.com
samankwatia@marburylaw.com

Date: December 21, 2023

KING & SPALDING LLP

BY: /s/ Brian Eutermoser
Brian Eutermoser
Registration No. 64,058

Attorney for Patent Owner *Open Text Inc.*