Trials@uspto.gov
571-272-7822

Paper 8
Date: February 7, 2024

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AO KASPERSKY LAB,
Petitioner,

v.

OPEN TEXT INC.,
Patent Owner.

IPR2023-01160
Patent 11,409,869 B2

Before JUSTIN T. ARBES, BARBARA A. PARVIS, and GARTH D. BAER, *Administrative Patent Judges.*

PARVIS, *Administrative Patent Judge*.

DECISION
Denying Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

AO Kaspersky Lab ("Petitioner") filed a Petition (Paper 1, "Pet.") requesting *inter partes* review of claims of claims 18–20, 22, 24–34, 36, 38–48, 50, and 52–59 ("challenged claims") of U.S. Patent No. 11,409,869 B2 (Ex. 1001, "the '869 patent").[1] Open Text Inc. ("Patent Owner") filed a Preliminary Response. Paper 7 ("Prelim. Resp.").

We have authority to determine whether to institute review under 35 U.S.C. § 314 and 37 C.F.R. § 42.4. We may institute an *inter partes* review if "the information presented in the petition . . . and any response . . . shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition." 35 U.S.C. § 314(a). For the reasons provided below, we determine that Petitioner has not demonstrated a reasonable likelihood that it will prevail in showing the unpatentability of at least one challenged claim. Accordingly, we do not institute *inter partes* review.

II. BACKGROUND

A. Real Parties-in-Interest

Petitioner identifies itself as the real party-in-interest. Pet. 2. Patent Owner indicates the "real parties-in-interest are Open Text Inc. ('Patent Owner') and Webroot LLC," and "further identifies the following affiliates: Carbonite, LLC; Open Text Corporation; and Open Text Holdings, Inc." Paper 4, 1.

[1] Although claim 50 is omitted from the list of challenged claims on page 1 of the Petition, the Petition includes claim 50 in the summary table of asserted grounds and provides an element-by-element analysis for the claim. Pet. 1, 10, 49–50, 53, 57. We treat the omission of claim 50 from page 1 of the Petition as a harmless typographical error.

B. Related Matters

Each party identifies a judicial matter that would affect, or be affected by, a decision in this proceeding. In particular, the parties inform us that the '869 patent is asserted in the following district court proceedings: *Webroot Inc, et al. v. Sophos Ltd.*, No. 6:22-cv-240 (W.D. Tex.); *Webroot, Inc. et al. v. CrowdStrike Holdings, Inc. et al.*, No. 6:22-cv-241; *Webroot, Inc. et al. v. AO Kaspersky Lab*, No. 6:22-cv-243[2]; and *Webroot, Inc. et al. v. Forcepoint LLC*, 6:22-cv-00342. Pet. 2–3; Paper 4, 1. Patent Owner also identifies *Webroot, Inc. et al. v. Trend Micro, Inc.*, No. 6:22-cv-239. Paper 4, 1. The parties also identify the following PTAB proceedings concerning the '869 patent's parent patent, U.S. Patent No. 10,599,844: IPR2023-00528 and IPR2023-00662. Pet. 2–3; Paper 4, 2–3. Patent Owner also identifies IPR2023-01053 and PGR2023-00031 that involve challenges to the '869 patent. Paper 4, 1–2.

C. The '869 Patent

The '869 patent is titled "Automatic Threat Detection of Executable Files Based on Static Data Analysis." Ex. 1001, code (54). The '869 patent issued from U.S. Application No. 16/791,649, filed on February 14, 2020 ("the '649 application"). Ex. 1001, codes (21), (22). The '649 application was filed as a continuation of U.S. Application No. 14/709,875, filed on May 12, 2015, which is now U.S. Patent No. 10,599,844 B2 ("the '844 patent"). *Id.*, code (63).

The '869 patent describes extracting "static data points" from the file and applying to them a classifier that was trained from known malicious,

[2] Petitioner indicates these cases "have been consolidated" as *Webroot, Inc. v. AO Kaspersky Lab*, No. 6:22-cv-00243-ADA-DTG (W.D. Tex.). Pet. 2.

benign, and unwanted executable files. *Id.* at 1:34–40. The general architecture of the system is shown in Figure 1, reproduced below:

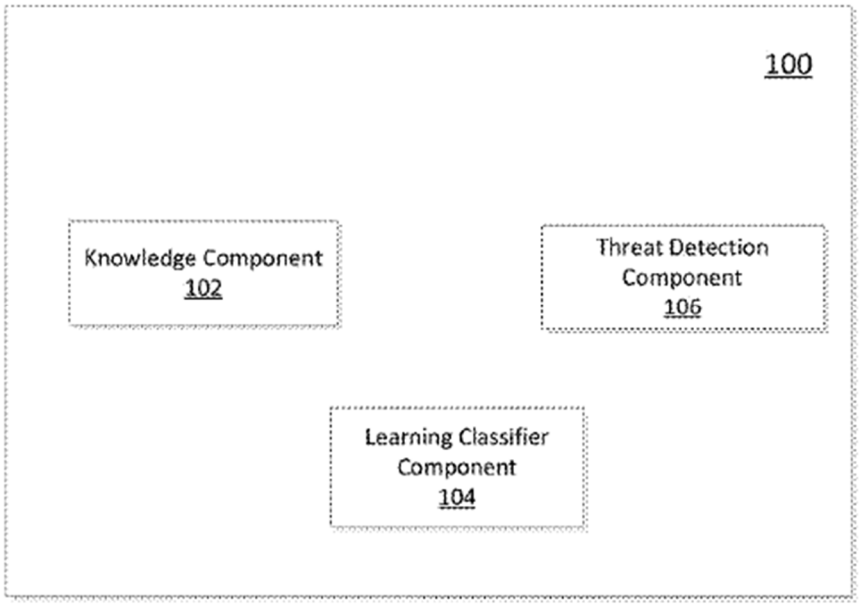


FIG. 1 illustrates exemplary system 100 showing components for threat detection. Id. at 1:59–61.

System 100 comprises knowledge component 102, learning classifier component 104, and threat determination component 106. *Id.* at 3:31–34. "To classify executable files . . . the knowledge component 102 collects static data on a large variety of executable files." *Id.* at 4:29–31. The data collected "yields a knowledgebase that may be used to periodically train a classifier, e.g. learning classifier component 104," which "may be used to classify an executable file as one of the following classifications: malicious, benign or potentially unwanted." *Id.* at 4:39–46. "Continuous update of the data maintained by the knowledge component 102 in conjunction with on-going re-learning/re-training of the learning classifier component 104 ensures that system 100 is up-to-date on current threats." *Id*. at 4:63–67.

The collected data "is analyzed to identify static data points that may

indicate one of a malicious file, a benign file or a potentially unwanted file." Ex. 1001, 5:8–11. Static data points "include, but are not limited to: header information, section information, import and export information, certificate information, resource information, string and flag information, legal information, comments, and/or program information (e.g., APIs and DLLs), among other examples." *Id.* at 5:62–67. Table 1.1 of the '869 patent, showing possible static data points, is reproduced below:

TABLE 1.1

Numeric values	Nominal values	Strings/Byte sequences	Boolean values
File size	initialize	Comments	Address Of Entry Point Anomaly
linker version	Un-initialize	company name	Image Base Anomaly
code size	entry point	file description	Section Alignment Anomaly
OS version	subsystem	internal name	Size Of Code Mismatch Anomaly
image version	file subtype	legal copyright	Low Import Count Anomaly
subsystem version	language	original file	Entry Point Anomaly
file version number	file flags masks	private build	certificate Validity
product version number	file flags	product name	Certificate Exception
size of heapr	file OS	special build	Code Characteristics Anomaly
size of stackr	file type	product version	Code Name Anomaly
size of image	machine type	file version	Count Anomaly
PE header time	PE type	package code	Data Characteristics Anomaly
Section Entropy	section counts	product code	Data Name Anomaly
Sections count	DLL count	export DLL name	Export Exception
	DLL functions	assembly version	Large Number of DLLs Anomaly
	data directory	Certificate Issuer	flag DLL Name Anomaly
	export count	Certificate Subject	Number of Functions Anomaly
	Earliest Data Byte	Imports	Function Name Anomaly
	resources	Exports	PE Header Anomaly
	resources language	Section Names	High Section Count Anomaly
	resource Encoding	Non-resource section strings	PE Magic Validity
	resource code page		Resource Exception
	resource size		VR Code Ratio Anomaly
	DLL characteristics		Import Exception

Table 1.1 shows possible static data points, organized by category.

The learning classifier component 104 "interfaces with the knowledge component 102 and a threat detection component 106 for the system 100 to

evaluate an executable file as a possible threat," such as by applying "programming operations or machine learning processing to evaluate static data points extracted from a file to be analyzed." Ex. 1001, 7:28–34. It "builds a learning classifier based on information from the knowledge component 102 including the extracted static data points of a file and the information used to train/re-train the learning classifier." *Id.* at 7:34–42.

"[T]he learning classifier can adaptively set features of a learning classifier based on the static data points extracted for a file." Ex. 1001, 7:43–45. For example, "ranges of data can be identified for static data points that may enable the learning classifier to controllably select (e.g., turn on/off) features for evaluation by learning classifier." *Id.* at 7:45–48. As a more specific example, "where a static data point extracted for evaluation is a file size . . . , the learning classifier may turn on/off features for evaluation based on whether . . . the file size . . . is within a certain range." *Id.* at 7:48–52.

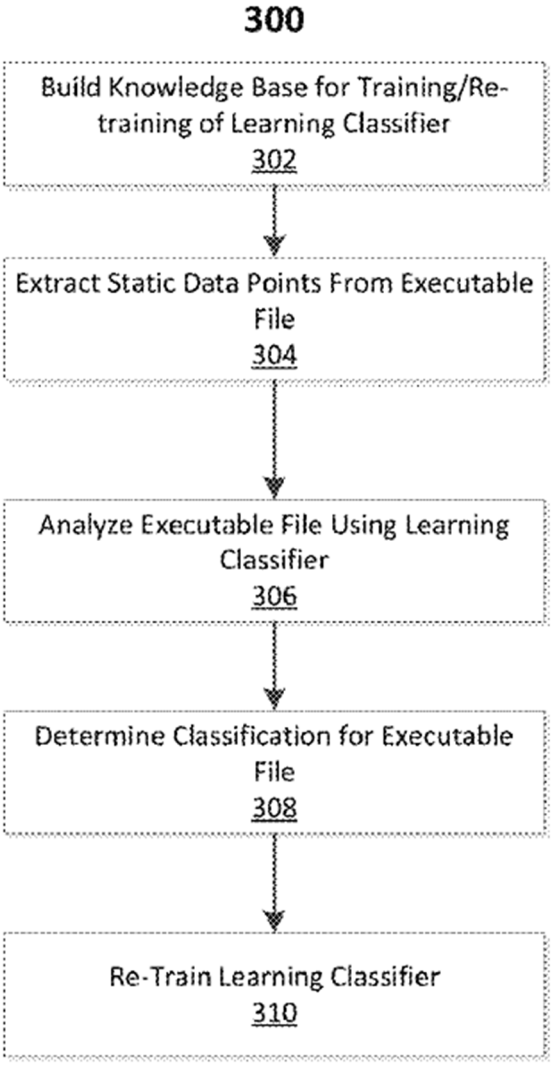
"The learning classifier component 104 processes and encodes collected/extracted data from a file to make the static data points suitable for processing operations" and "the different encoded data may be combined into one or more final feature vectors." *Id.* at 8:8–10, 8:31–34.

The learning classifier component 104 then "provides the one or more feature vectors as input to a support vector machine (SVM)," which "may then perform data analysis and pattern recognition on the one or more feature vectors" and "build a probabilistic model that indicates whether or not a file may be malicious." *Id.* at 9:27–34.

The threat detection component 106 "evaluates results of the processing performed by the learning classifier component 104 to make a determination as to whether the executable file is malicious, benign or a potentially unwanted file/application." *Id.* at 9:55–59. For example, "a final

score (e.g., probability value/evaluation) [may be] determined based on feature vector processing performed by the learning classifier component 104 and compared with a probability threshold value for determining whether an executable file is malicious." *Id.* at 10:3–28.

Figure 3, reproduced below, is a flowchart showing exemplary threat detection method 300:



"FIG. 3 illustrates an exemplary method 300 for implementation of threat detection systems and methods." Ex. 1001, 1:65–67.

The method begins at "operation 302 where a knowledge base is built

for training/retraining of a learning classifier used to detect threats in executable files . . . from data collected and evaluated related to known malicious executable files, known benign executable files and potentially unwanted executable files" by knowledge component 102. *Id.* at 11:60–67. The knowledge base "may be used to automatically train/re-train one or more learning classifiers used to evaluate threats in executable files based on the known malicious executable files, known benign executable files and the potentially unwanted executable files." *Id.* at 11:67–12:4.

"When an executable file is identified for evaluation," the method proceeds to operation 304, "where a one or more static data points are extracted from [the] executable file" and the file "is analyzed using machine learning processing as described with respect to the knowledge component 102 . . . to collect/extract static data points . . . for evaluation." Ex. 1001, 12:11–17. This may happen with or without decrypting or unpacking the file. *Id.* at 12:17–19.

At operation 306, "the executable file is analyzed for threats" by "applying a learning classifier (e.g., the learning classifier generated during performance of operation 302) to the plurality of static data points extracted from the file." Ex. 1001, 12:43–51. The system may "generat[e] at least one feature vector from the plurality of static data points extracted using the learning classifier trained by the knowledge base" and the "data may be parsed and encoded for machine learning processing." *Id.* at 12:55–60.

As noted above, "generation of the feature vector may comprise selectively setting features of the learning classifier based on the one or more of [the] static data points extracted" and "[f]eatures of the generated feature vector may be weighted based on classified categories identified by the knowledge base . . . and the plurality of static data points extracted from

the file." *Id.* at 12:61–67. In addition, "one or more features of the feature vector may be selectively turned on or off based on evaluation of whether a value of a static data point is within a predetermined range." *Id.* at 12:67–13:3. The feature vector is evaluated "using linear or nonlinear support vector processing to determine a classification for the executable file, for example whether the executable file is harmful, benign, or unwanted." *Id.* at 13:7–11.

At operation 308, "a determination is made as to a classification of the executable file," i.e., "whether the executable file is harmful (e.g., malicious, malware) or not based on results of the analysis of the executable file (e.g., using machine learning processing by a learning classifier)." Ex. 1001, 13:12–18. The "results of the analysis of [the] executable file may be data obtained from learning classifier, such as, for example an SVM, processing data." *Id.* at 13:18–20. The classification "may be based on any type of analytical, statistical or graphical analysis, or machine learning processing." *Id.* at 13:32–35.

"At any point in time, operation 310 may occur where a learning classifier used for threat detection processing is re-trained" to "ensure that the threat detection application/service is up to date and able to accurately detect new threats." *Id.* at 13:40–42.

D. Illustrative Claim

Petitioner challenges claims 18–20, 22, 24–34, 36, 38–48, 50, and 52–59 of the '869 patent. Pet. 1. Claims 18, 32, and 46 are independent. Independent claim 18, reproduced below[3], is illustrative of the claimed subject matter.

[3] We use Petitioner's annotations. Pet. 18–30.

[18.0] A computer-implemented method comprising:

[18.1] identifying static data points that may be indicative of either a harmful or benign executable file;

[18.2] associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;

[18.3] identifying an executable file to be evaluated;

[18.4] extracting a plurality of static data points from the executable file;

[18.5.a] generating a feature vector from the plurality of static data points [18.5.b] using a classifier trained to classify the static data points based on training data, [18.5.c] the training data comprising files known to fit into one of the plurality of categories of files, [18.5.d] wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and

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

Ex. 1001, 17:61–18:14.

E. Evidence

Petitioner relies on the patent document references summarized in the table below.

Name	Patent Document	Exhibit
Ideses	US 2015/0213376 A1	1005
Sharma	US 2016/0154960 A1	1006

Petitioner also relies on the Declaration of Venkatramanan Siva Subrahmanian, Ph.D. (Ex. 1002) to support its contentions that the challenged claims are unpatentable. Patent Owner relies on the Declaration of Sam Malek, Ph.D. (Ex. 2001).

F. Asserted Grounds

Petitioner asserts that the challenged claims of the '869 patent are unpatentable based on the following grounds summarized in the table below:

Claims Challenged	35 U.S.C. §[4]	References/Basis
18, 20, 22, 24–32, 34, 36, 38–46, 48, 50, 52–59	103	Ideses
18–20, 22, 24–34, 36, 38–48, 50, 52–59	103	Ideses, Sharma

III. ANALYSIS

A. Legal Standards

A patent claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time of the effective filing date of the claimed invention to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) when in evidence, objective evidence of nonobviousness, i.e., secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

B. Level of Ordinary Skill in the Art

Petitioner asserts the following:

[4] Because the challenged claims of the '869 patent have an apparent effective filing date after March 16, 2013, the 35 U.S.C. § 103 provision of the Leahy-Smith America Invents Act ("AIA"), Pub. L. No. 112-29, §§ 3(b)–3(c), 3(n)(1), 125 Stat. 284, 285–87, 293 (2011) applies.

> A Person of Ordinary Skill in The Art ("POSITA") in May 2015 (the earliest alleged priority date) would hold a bachelor's degree in computer science or a similar field, have about 2 years of experience as a software engineer or similar title, and have about 2 years of experience specifically in the computer security field. A person having more education, such as a master's degree, would require less experience, and a person not having at least a bachelor's degree would need more experience.

Pet. 8 (citing Ex. 1002 ¶¶ 12–13). At this stage in the proceeding, Patent Owner does not dispute Petitioner's proposal. Prelim. Resp. 31.

Considering the subject matter of the '869 patent, the background technical field, and the asserted prior art, we agree with Petitioner's proposed qualifications for an ordinary level of skill. Petitioner's proposed definition is consistent with the level of skill reflected in the specification of the '869 patent and the asserted prior art references. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001).

Accordingly, for purposes of this Decision, and based on the current record, we adopt Petitioner's assessment of the level of skill for one of ordinary skill in the art.[5]

C. Claim Construction

We interpret the challenged claims

> using the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.

[5] A similar, but slightly different level of skill was adopted in the related PGR2023-00031 Decision Denying Institution. *See* PGR2023-00031, Paper 7, 13. Our analysis and conclusions herein would be the same if we adopted the level of skill we adopted in PGR2023-00031.

37 C.F.R. § 42.100(b). Under this standard, the words of a claim generally are given their "ordinary and customary meaning," which is the meaning the term would have to a person of ordinary skill at the time of the invention, in the context of the entire patent including the specification. *Phillips v. AWH Corp*., 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc).

Petitioner states in the "Parallel Litigation, the Court construed the following terms from the '844 patent" and then Petitioner provides the claim terms and the Court's construction. Pet. 8–9. Petitioner also states that the "parties did not re-brief these terms for the '869 patent, rather the parties stipulated to the district court's constructions for the same terms from the '844 patent while reserving appellate rights." *Id.* at 9–10. Petitioner further states for purposes of this decision, "the Board need not expressly construe any additional terms." *Id.* at 10. Patent Owner indicates the "parties agree that no claim terms need be construed for this Petition." Prelim. Resp. 30. We determine no claim term needs to be construed expressly to resolve whether to institute *inter partes* review. *See Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019) ("The Board is required to construe 'only those terms . . . that are in controversy, and only to the extent necessary to resolve the controversy.'" (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc*., 200 F.3d 795, 803 (Fed. Cir. 1999))).

D. Unpatentability of Claims under 35 U.S.C. § 103 as Obvious over Ideses

Petitioner asserts that claims 18, 20, 22, 24–32, 34, 36, 38–46, 48, 50, and 52–59 are unpatentable as obvious over Ideses. Pet. 18–52. Patent Owner disputes Petitioner's assertions. Prelim. Resp. 32–41.

We begin with an overview of Ideses. We then turn to the parties' contentions.

1. Ideses

Ideses is titled "Methods and Systems for Generating Classifiers for Software Applications." Ex. 1005, code (54). Ideses "relates to . . . methods and systems for generating classifiers for software applications and, more specifically . . . to methods and systems for generating classifiers for software applications based on large feature vectors." *Id.* ¶ 2.

Figure 1 of Ideses is reproduced below.

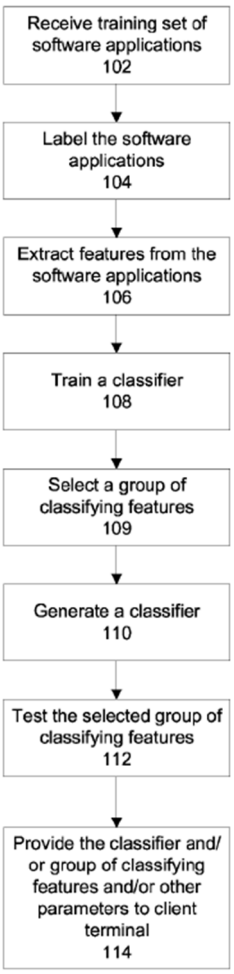


Figure 1 is a "flowchart of a method of generating a pruned feature list for classification." Id. ¶ 39.

At step 102, the system receives multiple software applications for training a classifier and, at step 104, the applications may be "labeled with a

classification type based on desired software application classification categories." *See id.* ¶¶ 80–81.

At step 106, features may be extracted from the software applications by a feature extractor module and "[o]ptionally, a feature vector [may be] extracted." *See id.* ¶ 86. The extracted features may be any feature that describes the software application, including static meta data, information regarding the executable code, or behavioral features. *See id.* ¶ 89. The features may be stored as feature vectors, which may be used for training a classifier. *See id.* ¶ 91.

At step 108, "a classifier [may be] trained based on the set of extracted features (block 106) and software classification labeling (block 104)." *Id.* ¶ 94. At step 109, "a group of classifying features is selected, for example, by a combination and/or selection module 306." *See id.* ¶ 98. "[E]xtracted features [may be] identified for pruning to generate the group of classifying features," and "the selection [may be] based on identifying a classification effectiveness rank for each of the features of the training software applications" or "based on identifying resource requirements for each of the features of the training software applications." *See id*. ¶¶ 98, 99.

The selection module may use "the parameters and/or coefficients gathered during the classification training" and "the group of classifying features [may be] selected based on the significance of the extracted features," such that "[s]ignificant features, such as those that have a large effect on the classification outcomes, may be retained" and "[i]nsignificant features, such as those that have a small, negligible or no effect on classification outcome may be pruned." *Id.* ¶¶ 101, 106.

"[F]eatures (or their equivalents) that do not contribute (or do not significantly contribute) to the classification process [may be] pruned or not

included," "features that correspond to coefficients having a value of zero [may be] pruned or not included, for example, when classification is performed based on a support vector machine (SVM)," and "features with a low magnitude value of the coefficient [may be] identified for pruning and/or are not included." *Id.* ¶ 107.

At step 110, "a classifier is generated, for example, by a classifier generating module" that may be "based on the group of classifying features." *Id.* ¶¶ 115–116. The feature set used to generate the trained classifier may be pruned. *See id*. ¶¶ 116–117.

Step 112 is an optional step of evaluating the "classifier performance based on the pruned feature set [as] compared to classifier performance based on the complete feature set" and "[t]he pruning may be selectively adjusted based on the testing." *See id*. ¶¶ 118–119.

Finally, at step 114, "the group of classifying features, the pruned feature vector, selected coefficients, and/or trained classifier are provided to [a] mobile device" "for run-time classification of software applications by the mobile device." *See id*. ¶¶ 120–122.

2. Analysis

The parties dispute whether Ideses teaches or suggests turning features on or off at the classification stage based on the executable being classified at that stage. *See* Pet. 18–52; Prelim. Resp. 32–41. The dispute, more specifically, pertains to claim elements 18.5.a and 18.5.d, which recite "generating a feature vector from the plurality of static data points," "wherein one or more *features of the feature vector are selectively turned on or off*" (emphasis added). Independent claims 32 and 46 have the same recitations.

For element 18.5.a, which recites "generating a feature vector from the plurality of static data points," Petitioner argues

> First, Ideses discloses "generating a selected group of classifying features (e.g., feature vector) for classification of software applications." Ex. 1005, ¶0071. Second, Ideses discloses the extracted features (e.g., static data points) are collated into feature vectors. Ex. 1005, ¶0134; *see also* [18.4]; Ex. 1002, ¶¶63, 143-144.

Pet. 24 (emphasis omitted).

For element 18.5.d, which recites, "wherein one or more features of the feature vector are selectively turned on or off," Petitioner directly quotes four sentences of Ideses, including two sentences in paragraph 44 and one sentence in each of paragraphs 71 and 107, each of which relate to generating classifiers. *Id.* at 27–28 (citing, e.g., Ex. 1005 ¶¶ 44, 71, 107). In particular, in paragraph 44, Ideses describes "selecting a set of classifying features extractable from software applications" "in order to *generate* classifiers." Ex. 1005 ¶ 44 (emphasis added). In paragraph 71, Ideses references Figure 1 and describes "[a]lternatively or additionally, each feature of the group is selected from-the-ground-up; added to the group as opposed to removed from or retained within [the] larger group." *Id.* ¶ 71. Figure 1 of Ideses describes a method for *generating* a classifier. *Id.* at Fig. 1 (including step 110, which is "[g]enerate a classifier"). In paragraph 107, Ideses describes "features (or their equivalents) that do not contribute (or do not significantly contribute) to the classification process are pruned or not included." *Id.* ¶ 107. The pruning described in paragraph 107 is performed to generate a classifier in accordance with the method of Figure 1. *Id.* ¶ 98 (describing "pruning to generate the group of classifying features" that occurs "[a]t 109" of Figure 1).

Patent Owner argues that in Ideses, "at the *training* stage, the system selects the features that the classifier will be trained to consider" and, thus "at the later classification stage, the set of features that must be extracted and considered has already been predetermined, such that there is *no feature selection at the classification stage*." Prelim. Resp. 36. Patent Owner argues in contrast, "the idea claimed in the '869 patent is to turn features on or off at the *classification* stage, based on the executable being classified at *that* stage." *Id.* at 36–37.

Patent Owner argues that "language of the independent claims clearly requires turning features on or off at the *classification stage* because the generated feature vector—whose features are 'turned on or off'—is not generated until that stage." *Id.* at 37. According to Patent Owner, "[b]ecause the recited turning of features 'on or off' is performed on the feature vector created at the classification stage, that turning on or off necessarily happens at the classification stage too." *Id.*

Patent Owner argues that other claim language supports its position including that "the feature vector whose features are being turned on or off is generated 'using a classifier [that has already been] ***trained***'—past tense." *Id.* Patent Owner also argues that its position is supported by an additional feature recited in independent claims 18, 32, and 46, specifically, "that the feature vector is used to classify the identified executable." *Id.* Patent Owner also argues that further support is found in the recitation of training and classification steps separately in claim 20. *Id.* at 37–38.

Patent Owner asserts that "[t]he benefit of turning features on or off at the classification stage is that classifier performance can be optimized *based on the executable being classified*" and "the classifier is not locked into evaluating the full set of features that was predetermined at the training stage

without regard to the executable being classified at the classification stage." *Id.* at 38.

We agree with Patent Owner that the claims are directed to the classification stage, as they require extracting "a plurality of static data points" from an executable file and "generating a feature vector from [that] plurality of static data points *using a classifier*" (emphasis added). The claims also recite that the classifier has been "*trained* to classify" (emphasis added). Further, the claims recite that "one or more features of the feature vector" are selectively turned on or off, indicating that the feature vector has been generated.

We also agree with Patent Owner that Petitioner has not shown sufficiently for institution that Ideses teaches or suggests a classification method in which one or more features of a feature vector generated from a plurality of static data points are selectively turned on or off based on the value of one of the static data points. As explained in the summary of Ideses above, Ideses describes "methods and systems for generating classifiers" (Ex. 1005 ¶ 2) in which "extracted features [may be] identified for pruning to generate the group of classifying features," and "the selection [may be] based on identifying a classification effectiveness rank for each of the features of the training software applications" or "based on identifying resource requirements for each of the features of the training software applications." *See id*. ¶¶ 98, 99. "[T]he group of classifying features [may be] selected based on the significance of the extracted features," such that "[s]ignificant features, such as those that have a large effect on the classification outcomes, may be retained" and "[i]nsignificant features, such as those that have a small, negligible or no effect on classification outcome may be pruned." *Id.* ¶¶ 101, 106.

Importantly, all the selection and pruning described in Ideses concerns the creation of the classifier, where the features that are selected or pruned are either included or omitted from the classifier. Ideses's Figure 1 depicts a flowchart in which a classifier is *generated* at step 110 and the classifier is provided to a client terminal at step 114, which is the last step of the method shown in Figure 1. Ex. 1005, Fig. 1. Ideses's Figure 2 depicts a method of classification based on the pruned feature list. *Id.* ¶ 40. Ideses's Figure 3 depicts central processor 302 and mobile device 308, which communicate with each other through network 316. *Id.* ¶ 77. Ideses describes that "central processor 302 generates the selected group of classifying features, the selected coefficients, and/or the trained classifier" and the "selected group of classifying features and/or the trained classifier is then provided to client devices 308 for local run-time classification." *Id.* ¶ 78.

For element 18.5.a, Petitioner argues that in Ideses, first, a selected group of classifying features is generated. Pet. 24 (citing Ex. 1005 ¶ 71). That disclosure pertains to Figure 1. Ex. 1005 ¶ 71. Petitioner argues second in Ideses, "the extracted features (e.g., static data points) are collated into feature vectors." Pet. 24 (citing Ex. 1005 ¶ 134) (emphasis omitted). That disclosure pertains to Figure 2. Ex. 1005 ¶ 134. Petitioner also relies on its arguments for element 18.4. Pet. 24 ("*see also* [18.4]"). Petitioner's arguments relating to element 18.4 pertain to whether Ideses discloses "static data points." *Id.* at 22–24.

For element 18.5.d, which requires that one or more features of the feature vector "are selectively turned on or off" during the classification stage, as discussed above, Petitioner relies on Ideses's disclosures of selection and pruning that are described expressly with respect to *generating* classifiers. *See* Pet. 27–28; Ex. 1005 ¶¶ 44 (describing selecting a set of

classifying features to "generate classifiers"), 71 (describing generating a selected group of classifying features in accordance with the method shown in Figure 1), 98, 107 (describing with respect to step 109 shown in Figure 1 pruning features that do not significantly contribute to generate the group of classifying features). Petitioner does not rely on disclosures relating to Figure 2 for element 18.5.d. Pet. 27–28.

Petitioner fails to show how the disclosures relied on in Ideses teach or suggest "generating a feature vector from [a] plurality of static data points using a classifier" where features are "selectively turned on or off" based on a value of one of the static data points. Even if Ideses's omission of insignificant features could be considered "turn[ing] [them] on or off," that happens in the *creation* of the classifier. Ideses does not teach or suggest "wherein one or more *features of the feature vector* are selectively turned on or off" subsequent to "generating a *feature vector* . . . using *a classifier*," as recited in each challenged claim.

Petitioner also relies on the testimony of Dr. Subrahmanian. Pet. 27–28 (citing Ex. 1002 ¶¶ 164–168). Dr. Subrahmanian testifies that "much research has gone into identifying the best subset of features from a larger set of features" and cites to Exhibit 1028 as supporting evidence. Ex. 1002 ¶ 166. Dr. Subrahmanian also testifies that "[a]dding new features to a given suite of features has been a common aspect of machine learning research before the priority date of the '869 patent" and cites to Exhibit 1029 as supporting evidence. *Id.* ¶ 167. We find that Dr. Subrahmanian's testimony (*id.* ¶¶ 166–167) simply provides additional evidence that Ideses's teachings would have been known to a person having ordinary skill in the art. Our finding is consistent with Petitioner's argument that its "analysis . . . cites additional prior art to demonstrate the background knowledge of a POSITA

and to provide contemporaneous context to support Petitioner's assertions regarding what a POSITA would have understood from the prior art." Pet. 11.

The Petition includes no arguments similar to Dr. Subrahmanian's testimony in connection with element 18.5.d. Pet. 27–28. Instead, for element 18.5.d, the Petition includes a statement that Ideses "discloses" element 18.5.d and quotes four sentences of Ideses. *Id.* Additionally, Exhibits 1028 and 1029 are not raised anywhere in the Petition. *Id.* at 2 (not including either Exhibits 1028 or 1029 in the unpatentability grounds), 11–18 (providing overviews of only Ideses and Sharma), 18–31, 42, 48 (setting forth an element-by-element analysis based on the teachings of only Ideses). To the extent Petitioner intends to rely on Exhibits 1028 and 1029 to remedy the deficiencies in Ideses, Petitioner's analysis in that regard needed to be set forth in the Petition. *See* 37 C.F.R. § 42.6(a)(3) ("Arguments must not be incorporated by reference from one document into another document.").

Also, although Dr. Subrahmanian testifies regarding research for identifying the best subset of features from a larger set of features and adding features to a given set of features, based on Exhibits 1028 and 1029 that research relates to generating classifiers. *See* Ex. 1028, 182 (describing "feature reduction phase" of the experiment), 183 (describing subsequent "classification stage"); Ex. 1029, 5–6 (describing that feature extraction "was performed," the classifier was "optimized", and then the classifier was "run on the independent test data for analysis"). Petitioner has not shown sufficiently for institution that Ideses teaches or suggests "wherein one or more *features of the feature vector* are selectively turned on or off" subsequent to "generating a *feature vector* . . . using *a classifier*," as recited in each challenged claim. Dr. Subrahmanian does not testify regarding any

research or work involving identifying the best subset of features or adding new features to a given suite of features *during the classification stage*. Also, neither the Petition nor Dr. Subrahmanian's testimony includes a reason to modify the teachings of Ideses to address element 18.5.d. Ex. 1002 ¶¶ 164–168; Pet. 2, 11–18, 18–31, 42, 48. We, therefore, find that Dr. Subrahmanian's testimony does not remedy the deficiencies discussed with respect to arguments in the Petition regarding Ideses.

For these reasons, we find that Petitioner has not shown sufficiently for institution that Ideses teaches or suggests claim elements 18.5.a and 18.5.d, which recite "generating a feature vector from the plurality of static data points" and "wherein one or more features of the feature vector are selectively turned on or off." Petitioner has not remedied the deficiency in its arguments for independent claims 32 and 46, which have the same recitation. Petitioner also has not remedied the deficiency in its arguments for the dependent claims, each of which depends, directly or indirectly, from independent claim 18, 32, or 46.

Accordingly, Petitioner has not shown a reasonable likelihood of prevailing on its assertion that claims 18, 20, 22, 24–32, 34, 36, 38–46, 48, 50, and 52–59 are unpatentable as obvious over Ideses.

E. Unpatentability of Claims under 35 U.S.C. § 103 as Obvious over Ideses and Sharma

Petitioner asserts that claims 18–20, 22, 24–34, 36, 38–48, 50, and 52–59 are unpatentable as obvious over Ideses and Sharma. Pet. 53–63. Patent Owner disputes Petitioner's assertions. Prelim. Resp. 41–47.

We begin with an overview of Sharma. We then turn to the parties' contentions.

1. Sharma

Sharma is titled "Systems and Methods for Risk Rating Framework for Mobile Applications." Ex. 1006, code (54). Sharma "relates to methods, systems, and computer-readable medium for determining a risk-rating for computer executable applications." *Id.* ¶ 2.

Sharma describes a risk-rating framework for evaluating a risk or vulnerability of a mobile computer application before installing it on a user device. *Id.* ¶ 24. Sharma's risk-rating framework system is a machine learning system that uses one or more classifiers. *Id.* Sharma's classifiers are trained and tested using an input data set and construct a model that aids in evaluating a risk level of computer applications. *Id.* ¶ 25.

2. Analysis

Petitioner relies on the combination of Ideses and Sharma as an alternative unpatentability argument for element 18.5.b, which recites "using a classifier trained to classify the static data points based on training data." Pet. 57–59. Claims 32 and 46 have the same recitations. Petitioner provides reasons to combine the teachings of Ideses with Sharma's teachings relating to training and testing classifiers using input data. *Id.* at 53–57. Petitioner does not provide reasoning to modify Ideses to remedy the deficiency discussed with respect to element 18.5.d and obviousness over Ideses alone.

Patent Owner argues "Petitioner fails even to argue—let alone show—that Sharma remedies the deficiencies of Ideses described above." Prelim. Resp. 41. Patent Owner states "Petitioner argues that Sharma teaches a different limitation." *Id.* at 44. Patent Owner also argues that "the only feature selection that Sharma teaches occurs at the training stage, not at the classification stage, as claimed." *Id.* (citing Ex. 2001 ¶¶ 65–73).

We agree with Patent Owner that Petitioner relies on the combination of Ideses and Sharma for a different limitation, which is "using a classifier trained to classify the static data points based on training data." Also, Sharma's disclosures relied on by Petitioner do not describe selectively turned on or off features at the classification stage. Instead, Sharma describes that the classifiers used in Sharma's risk-rating framework are "first trained and tested using an input data set that includes known malware and known-benign computer applications." Ex. 1006 ¶¶ 24–25 (cited in Pet. 57–58). In that regard, in the disclosures relied on by Petitioner, Sharma describes that "all . . . features are inputted to the classifier" and the "classifier is designed to consider each feature." *Id.* ¶ 47 (cited in Pet. 58). Petitioner has not shown sufficiently for institution that Ideses alone or in combination with Sharma teaches or suggests "generating a feature vector from the plurality of static data points" and "wherein one or more features of the feature vector are selectively turned on or off," recited in each of the independent claims. Accordingly, Petitioner has not shown a reasonable likelihood of prevailing on its assertion that claims 18–20, 22, 24–34, 36, 38–48, 50, and 52–59 are unpatentable as obvious over Ideses and Sharma.

IV. CONCLUSION

Based on the arguments and evidence presented by the parties, we conclude that Petitioner has not demonstrated a reasonable likelihood of prevailing with respect to at least one claim of the '869 patent challenged in the Petition. Therefore, we do not institute an *inter partes* review.[6]

[6] Because we deny the Petition on the merits, we do not reach Patent Owner's request that we exercise our discretion to deny the Petition based on the state of the related litigation. *See* Prelim. Resp. 3–19.

V. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that the Petition is denied and no *inter partes* review is instituted.

For PETITIONER:

Joseph T. Miotke
Eric Chadwick
Erin Block
DEWITT LLP
jtm@dewittllp.com
ehc@dewittllp.com
eblock@dewittllp.com
docketing@dewittllp.com

FOR PATENT OWNER:

Brian Eutermoser
Russell E. Blythe
Mikaela Stone
KING & SPALDING LLP
beutermoser@kslaw.com
rblythe@kslaw.com
mikaela.stone@kslaw.com
opentext-ipr-pgr@kslaw.com