

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AO KASPERSKY LAB
Petitioner

v.

WEBROOT, INC.
Patent Owner

IPR2023-_____
U.S. Patent No. 11,409,869

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

PETITIONER’S EXHIBIT LIST	v
I. INTRODUCTION	1
II. MANDATORY NOTICES.....	2
A. Real Party-in-Interest.....	2
B. Related Matters.....	2
C. Counsel.....	3
D. Service Information	3
III. CERTIFICATION OF GROUNDS FOR STANDING	4
IV. THE ‘869 PATENT	4
A. Overview of the ‘869 Patent	4
B. Prosecution History	6
V. LEVEL OF ORDINARY SKILL IN THE ART	8
VI. CLAIM CONSTRUCTION.....	8
VII. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED	10
VIII. PRIOR ART OVERVIEW	11
A. Ideses.....	11
B. Sharma	17
IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE	18
A. Ground 1: Claims 18, 20, 22, 24-32, 34, 36, 38-46, 48, 50, 52-59 are obvious under 35 U.S.C. § 103 by Ideses	18
1. Claim 18	18
2. Claim 20	31
3. Claim 22	32
4. Claim 24	33
5. Claim 25	34
6. Claim 26	34
7. Claim 27	35

8. Claim 28.....	36
9. Claim 29.....	36
10. Claim 30.....	37
11. Claim 31.....	39
12. Claim 32.....	40
13. Claim 34.....	43
14. Claim 36.....	43
15. Claim 38.....	43
16. Claim 39.....	44
17. Claim 40.....	44
18. Claim 41.....	44
19. Claim 42.....	45
20. Claim 43.....	45
21. Claim 44.....	45
22. Claim 45.....	46
23. Claim 46.....	46
24. Claim 48.....	49
25. Claim 50.....	49
26. Claim 52.....	50
27. Claim 53.....	50
28. Claim 54.....	50
29. Claim 55.....	51
30. Claim 56.....	51
31. Claim 57.....	51
32. Claim 58.....	52
33. Claim 59.....	52
B. Ground 2: Claims 18-20, 22, 24-34, 36, 38-48, 50, 52-59 are obvious under 35 U.S.C. § 103 over Ideses in view of Sharma	53
1. Reasons to Combine Ideses and Sharma	53

2. Elements [18.5.b], [32.5.b] and [46.5.b] (as recited above in Section IX.A.)	57
3. Claim 19	59
4. Claim 33	62
5. Claim 47	63
X. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE	63
A. THE BOARD SHOULD NOT DENY INSTITUTION UNDER SECTION 314(a)	63
1. Factor 1: Whether the district court granted a stay or a stay may be granted if a proceeding is instituted	63
2. Factor 2: Proximity of the court’s trial date to the Board’s projected statutory deadline for a final written decision	64
3. Factor 3: Investment in the parallel proceeding by the court and the parties	65
4. Factor 4: Overlap between issues raised in the petition and in the parallel proceeding	66
5. Factor 5: Whether the petitioner and the defendant in the parallel proceeding are the same party	67
6. Factor 6: Other circumstances that impact the Board’s exercise of discretion, including the merits	67
B. THE BOARD SHOULD NOT DENY INSTITUTION UNDER GENERAL PLASTIC.....	68
1. Factor 1: Whether the Same Petitioner Previously Filed a Petition	68
2. Factor 2: Whether at the Time of Filing of the First Petition the Petitioner Knew of the Prior Art Asserted in the Second Petition or Should Have Known of It.....	70
3. Factor 3: Whether at the Time of Filing of the Second Petition the Petitioner Already Received the Patent Owner’s Preliminary Response to the First Petition	71
4. Factors 4 & 5: The Length of Time that Elapsed between the Time the Petitioner Learned of the Prior Art Asserted in the Second Petition and the Filing of the Second Petition, and Whether the Petitioner Provides Adequate Explanation.....	72

5. Factors 6 & 7: The Finite Resources of the Board, and Requirement to Issue a Final Determination within One Year from Institution	72
C. THE BOARD SHOULD NOT DENY INSTITUTION UNDER 35 U.S.C. § 325(d)	73
1. <i>Advanced Bionics</i> Step 1 (<i>Becton</i> Factors (a), (b), (d)) favors institution ..	73
2. <i>Advanced Bionics</i> Step 2 (<i>Becton</i> Factors (c), (e), (f)) favors institution ...	74
XI. CONCLUSION	77
CERTIFICATE OF COMPLIANCE UNDER 37 C.F.R. § 42.24(d)	78
CERTIFICATE OF SERVICE	79

PETITIONER’S EXHIBIT LIST

Exhibit	Description
1001	U.S. Patent No. 11,409,869 (issued Aug. 9, 2022)
1002	Declaration of V.S. Subrahmanian, Ph.D. under 37 C.F.R. §1.68
1003	Curriculum Vitae of V.S. Subrahmanian, Ph.D.
1004	File Hist. U.S. Patent No. 11,409,869
1005	U.S. Patent Publication No. 2015/0213376 to Ideses et al. (published Jul. 30, 2015) (“Ideses”)
1006	U.S. Patent Application Publication No. 2016/0154960 to Sharma et al. (published Jun. 2, 2016) (“Sharma”)
1007	CHARLES LEDOUX & ARUN LAKHOTIA, <i>Malware and Machine Learning, in</i> INTELLIGENT METHODS CYBER WARFARE 1 (2014)
1008	U.S. Patent Application Publication No. 2015/0213365 to Ideses et al. (“Ideses ‘365”)
1009	Claim Construction Order, Mar. 16, 2023, Case No. 6:22-CV-00243-ADA-DTG
1010	Joint Stipulation Reserving Appellate Rights, Apr. 12, 2023, Case No. 6:22-CV-00243-ADA-DTG
1011	Kris Kendall, <i>Practical Malware Analysis, in</i> BLACK HAT CONF., USA (August 2007)
1012	Hao Helen Zhang et al., <i>Compactly Supported Radial Basis Function Kernels</i> , INST. STAT. MIMEO SERIES NO. 2570, N. CAROLINA ST. U. DEP’T. STAT. 2 (2004)
1013	Zane Markel & Michael Bilzor, <i>Building a Machine Learning Classifier for Malware Detection</i> , 2014 SECOND WORKSHOP ANTI-MALWARE TESTING RES. (WATeR) (2014)
1014	Naser Peiravian & Xingquan Zhu, <i>Machine Learning for Android Malware Detection Using Permission and API Calls, in</i> 2013 IEEE 25 TH INT’L CONF. TOOLS ARTIFICIAL INTELLIGENCE 300 (2013)
1015	Ekta Gandotra et al., <i>Malware Analysis and Classification: A Survey</i> , J. INF. SECURITY 5, 56-64 (2014)

Exhibit	Description
1016	Rafiqul Islam et al., <i>Classification of Malware Based on Integrated Static and Dynamic Features</i> , 36 J. NETWORK COMPUTER APPLICATIONS, 646 (2012)
1017	D. Michael Cai et al., <i>Comparison of feature selection and classification algorithms in identifying malicious executables</i> , 51(6) COMPUTATIONAL STAT. DATA ANALYSIS 3156 (2007)
1018	Guanhua Yan et al., <i>Exploring Discriminatory Features for Automated Malware Classification</i> , 10 DETECTION INTRUSIONS MALWARE, VULNERABILITY ASSESSMENT: 10 TH INT’L CONF., 41 (2013)
1019	Rafiqul Islam et al., <i>Classification of Malware Based on String and Function Feature Selection</i> , 2010 SECOND CYBERCRIME TRUSTWORTHY COMPUTING WORKSHOP 9 (2010)
1020	Eitan Menahem et al., <i>Improving Malware Detection by Applying Multi-Inducer Ensemble</i> , COMPUTATIONAL STAT. DATA ANALYSIS, 53(4), 1483 (2008)
1021	Ianir Ideses & Assaf Neuberger, <i>Adware Detection and Privacy Control in Mobile Devices</i> , in 2014 IEEE 28 TH CONVENTION ELECTRICAL ELECTRONICS ENGINEERS ISRAEL 1 (2014)
1022	Raja Khurram Shahzad et al., 2011, <i>Accurate Adware Detection Using Opcode Sequence Extraction</i> , in 2011 SIXTH INT’L CONF. AVAILABILITY, RELIABILITY SECURITY 189 (2011)
1023	Raymond Canzanese et al., <i>Toward an Automatic, Online Behavioral Malware Classification System</i> , in 2013 IEEE 7TH INT’L CONF. SELF-ADAPTIVE SELF-ORGANIZING SYS., 111 (2013)
1024	Yanfang Ye et al., <i>Automatic Malware Categorization Using Cluster Ensemble</i> , in PROC. 16TH ACM SIGKDD INT’L CONF. KNOWLEDGE DISCOVERY DATA MINING 95 (2010)
1025	Asaf Shabtai et al., <i>Detection of Malicious Code by Applying Machine Learning Classifiers on Static Features: A State-Of-The-Art Survey</i> , INFO. SECURITY TECH. REP. 14(1), 16 (2009)
1026	A.M. Aswini & P. Vinod, <i>Droid Permission Miner: Mining Prominent Permissions for Android Malware Analysis</i> , in FIFTH INT’L CONF. APPLICATIONS DIGITAL INF. WEB TECH. 81 (2014)

Exhibit	Description
1027	Konrad Rieck et al., <i>Learning and Classification of Malware Behavior</i> , in DETECTION INTRUSIONS MALWARE, VULNERABILITY ASSESSMENT: 5TH INT’L CONF., 108 (2008)
1028	Veelasha Moonsamy et al., <i>Feature Reduction to Speed Up Malware Classification</i> , in INF. SECURITY TECH. APPLICATIONS: 16TH NORDIC CONF. SECURE IT SYS., NORDSEC 2011, TALLINN, EST., OCT. 26-28, 2011, REV. SELECTED PAPERS 16, 176 (2012)
1029	John P. Dickerson et al., <i>Using Sentiment to Detect Bots on Twitter: Are Humans More Opinionated Than Bots?</i> , in 2014 IEEE/ACM INT’L CONF. ADVANCES SOCIAL NETWORKS ANALYSIS MINING 620 (2014)
1030	Wei Yu et al., <i>Towards Neural Network Based Malware Detection on Android Mobile Devices</i> , CYBERSECURITY SYS. HUMAN COGNITION AUGMENTATION 99 (2014)
1031	Guanhua Yan et al., 2013. <i>Exploring Discriminatory Features for Automated Malware Classification</i> , in DETECTION INTRUSIONS MALWARE, VULNERABILITY ASSESSMENT: 10TH INT’L CONF., DIMVA 2013, BERLIN, GER., JULY 18-19, 2013 PROC. 10, 41 (2013)
1032	Julia Yu-Cheng et al., <i>An Information Retrieval Approach for Malware Classification Based on Windows API Calls</i> , in 4 PROC. 2013 INT’L CONF. MACHINE LEARNING CYBERNETICS 1678 (2013)
1033	Zi Chu et al., <i>Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?</i> , 9 IEEE TRANSACTIONS DEPENDABLE SECURE COMPUTING 6, 811 (2012)
1034	Axelle Apvrille, <i>Android Reverse Engineering Tools From an Anti-Virus Analyst's Perspective</i> , slides 11-15 (March 2012), http://wikisec.free.fr/papers/insomnidroid.pdf
1035	Li Sun et al., <i>Pattern Recognition Techniques for the Classification of Malware Packers</i> , in 15 PROC. INF. SECURITY PRIV.: 15TH AUSTRALASIAN CONF., ACISP 2010, SYDNEY, AUSTL., JULY 5-7, 2010, 370 (2010)
1036	Munkhbayar Bat-Erdene et al., <i>Dynamic Classification of Packing Algorithms for Inspecting Executables Using Entropy Analysis</i> , in 2013 8TH INT’L CONF. MALICIOUS UNWANTED SOFTWARE: “AMERICAS” (MALWARE) 19 (2013)

Exhibit	Description
1037	M. Zubair et al., <i>PE-Probe: Leveraging Packer Detection and Structural Information to Detect Malicious Portable Executables</i> , in 8 PROC. VIRUS BULL. CONF. (2009)

I. INTRODUCTION

Pursuant to 35 U.S.C. §§ 311, 314(a), and 37 C.F.R. § 42.100, AO Kaspersky Lab (“Petitioner”) respectfully requests that the Board review and cancel as unpatentable claims 18-20, 22, 24-34, 36, 38-48, and 52-59 (the “challenged claims”) of U.S. Patent No. 11,409,869 (“‘869 patent,” Ex. 1001). The ‘869 patent describes systems and methods for threat detection of executable files using static data points extracted from the executable file without decrypting or unpacking the executable file. Ex. 1001, Abstract. The ‘869 patent claims systems and methods of using a feature vector for classifying the static data points and a machine learning model for modeling the executable for threat detection.

The systems and methods of threat detection claimed in the ‘869 patent were already well known when the inventors filed the application for the ‘869 patent. The prior art describes threat detection with these elements before the ‘869 patent priority date in May 2015. Most notably, Ideses et al. – filed in January 2015 and not considered during prosecution – discloses methods and systems for generating classifiers for software applications based on large feature vectors, including use of a support vector machine.

The evidence in this Petition demonstrates that the challenged claims are unpatentable over the prior art. Accordingly, the Board should institute review.

II. MANDATORY NOTICES

A. Real Party-in-Interest

The sole real party-in-interest for this Petition is AO Kaspersky Lab, Petitioner.

B. Related Matters

The Petitioner is the defendant in a case alleging infringement of the ‘869 patent styled *Webroot, Inc. et al. v. AO Kaspersky Lab* (Case No. 6:22-CV-00243-ADA-DTG), pending in the Western District of Texas, Waco Division.

The Petitioner further understands that the ‘869 patent has been asserted in the following matters, which matters have been consolidated only for pre-trial purposes with the above matter: *Webroot, Inc. et al. v. Sophos Ltd.* (Case No. 6:22-CV-00240-ADA-DTG); *Webroot, Inc. et al. v. CrowdStrike, Inc. et al.* (Case No. 6:22-CV-00241-ADA-DTG); and *Webroot, Inc. et al. v. Forcepoint, LLC* (Case No. 6:22-CV-00342-ADA-DTG) (collectively the “Parallel Litigation”). Each of the four cases will be tried separately.

Finally, the Petitioner understands that the ‘869 patent is the subject of the following PGR petition: *Sophos, Ltd. et al., v. Webroot, Inc. et al.* in PGR2023-0031 (“Sophos PGR”). Parent patent, U.S. Patent No. 10,599,844 is subject to the

following IPR petitions: *Sophos Inc. et al. v. Webroot, Inc. et al.* in IPR2023-00528 and *Trend Micro, Inc. v. Webroot, Inc. et al.* in IPR 2023-00662.

C. Counsel

Petitioner designates Joseph T. Miotke (Reg. No. 47,798) as lead counsel and Eric Chadwick (Reg. No. 41,664), and Erin Block (Reg. No. 78,154) as back-up counsel.

D. Service Information

Service of any documents may be made at the postal mailing address of Dewitt LLP, 901 Marquette Ave., 2100 AT&T Tower, Minneapolis, MN 55402, (Tel: (612) 305-1400 and Fax: (612) 305-1414).

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

jtm@dewittllp.com;

ehc@dewittllp.com;

eblock@dewittllp.com; and

docketing@dewittllp.com.

III. CERTIFICATION OF GROUNDS FOR STANDING

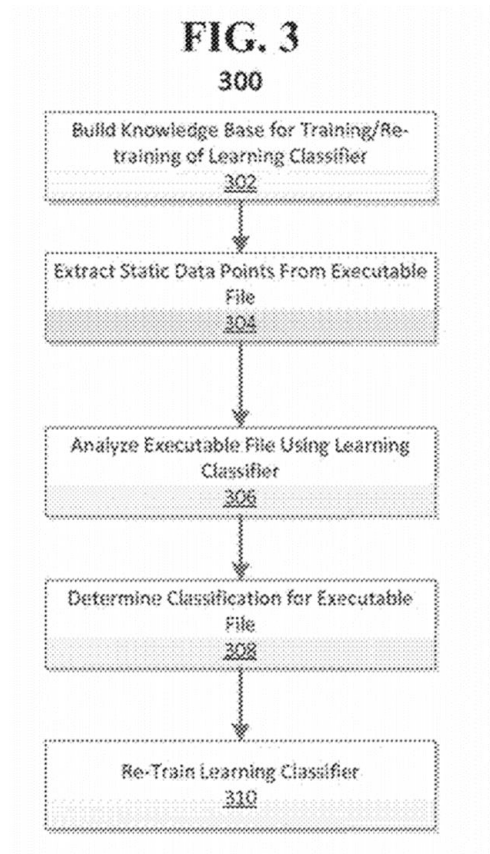
Petitioner certifies that the ‘869 patent is eligible for IPR, and that Petitioner is not barred or estopped from requesting IPR challenging the patent claims. 37 C.F.R. § 42.104(a).

IV. THE ‘869 PATENT

A. Overview of the ‘869 Patent

The ‘869 patent relates to training a classifier to determine whether an executable file is one of three classification types, harmful, benign, or potentially unwanted, using static data points (e.g., file size, code, size, product name) “before the executable is actually executed on a processing device.” Ex. 1001, Abstract, 1:31-32, 2:12-14, 4:43-46, Table 1.1. The patent discloses devices, methods, and systems for training a classifier and extracting static data points from an executable file, analyzing the executable file with the classifier, and determining the file’s classification type. Ex. 1001, Abstract, 1:31-45. The classifier “is trained from data comprising known malicious executable files, known benign executable files, and known unwanted executable files.” Ex. 1001, Abstract, 1:38-40.

Figure 3 depicts an “exemplary method” for performing threat detection:



Ex. 1001, FIG. 3. At operation 302, a knowledge base is built using known malicious executable files, known benign executable files, and known potentially unwanted executable files and used to train/re-train a classifier to evaluate threats in executable files. Ex. 1001, 11:60-12:4. At operation 304, when an executable file is identified for evaluation, static data points are extracted from an executable file. *Id.* at 12:11-13. At operation 306, the executable file is analyzed for threats. *Id.* at 12:43-44. At operation 308, a classification for the executable file is determined. *Id.* at 13:12-13.

B. Prosecution History

The ‘869 patent was filed as U.S. Application No. 16/791/649, (“the ‘649 application”), on February 14, 2020, as a continuation of U.S. Patent No. 10,599,844 (“the ‘844 patent”), filed May 12, 2015.

The ‘649 application received a number of Office Actions and Notices of Allowance. The Applicant reopened prosecution twice after notice of allowance (RCEs filed November 18, 2021, and May 10, 2022) to further amend the claims and add new claims.

In response to a Non-Final Rejection mailed March 29, 2021, in which allowable subject matter was indicated, Applicant amended the independent claims to include the requirement that “wherein one or more features of the feature vector are selectively turned on or off based on information associated with one or more of the plurality of extracted static data points,” in an Amendment filed June 17, 2021. Ex. 1004, 213. Applicant also filed a Terminal Disclaimer to overcome a double patenting rejection over the parent ‘844 patent. Ex. 1004, 196-197. A first Notice of Allowance was mailed August 18, 2021, in response to the Applicant’s amendments. Ex. 1004, 176.

The Applicant reopened prosecution to further amend the claims and add new claims and subsequently received a Non-Final Rejection mailed December 21, 2021,

in which allowable subject matter was indicated. In response, the Applicant further amended the independent claims to recite, in pertinent part, “wherein one or more features of the feature vector are selectively turned on or off ~~based on information associated with one or more of the plurality of extracted static data points~~ based at least in part on evaluation of whether a value of one or the plurality of static data points is within a predetermined range” in an Amendment filed February 8, 2022.

Ex. 1004, 118. A second Notice of Allowance was mailed April 15, 2022.

The Applicant again reopened prosecution to further add new claims and received a third Notice of Allowance on June 15, 2022. The Examiner concluded that the ’649 application was patentable over the prior art for the following reason: “the prior art, either alone or in a reasonable combination, fails to teach selectively turning on/off one or more features of the generated feature vector 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,” amendments that were made to the independent claims (or initially added to new independent claims). Ex. 1004, 23. The ’869 patent issued on August 9, 2022.

V. LEVEL OF ORDINARY SKILL IN THE ART

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. Ex. 1002, ¶¶ 12-13.

VI. CLAIM CONSTRUCTION

Claims subject to an IPR are construed using the same standards as in a civil action including giving claims their “ordinary and customary meaning” as understood by a skilled artisan and the prosecution history pertaining to the patent. 37 C.F.R. §42.100(b). In the Parallel Litigation, the Court construed the following terms from the ‘844 patent:

Term	Construction
“an executable file”	Plain-and-ordinary meaning

The terms relating to whether “character strings” are “within a predetermined range”	Not indefinite. Plain-and-ordinary meaning
“within a predetermined range”	Plain-and-ordinary meaning, wherein the plain-and-ordinary meaning of “range” is numerical range or a subset of a “particular scale” with at least a lower limit or an upper limit, but not necessarily both.
“support vector processing”	Plain-and-ordinary meaning wherein the plain-and-ordinary meaning is “processing that involves a support vector.”
“at least one processor . . .”	Plain-and-ordinary meaning. Not subject to §112(f)

Ex. 1009, 17-20. The above district court constructions are used in this Petition where applicable. 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. Ex. 1010.

Because a skilled artisan would find the challenged claims unpatentable under any interpretation consistent with their plain and ordinary meaning in the context of the '869 patent, the Board need not expressly construe any additional terms. *See Vivid Techs., Inc. v. Am. Sci. & Eng.'g. Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

VII. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED

Petitioner requests review under 35 U.S.C. § 311 and cancelation of claims 18-20, 22, 24-34, 36, 38-48, 50, and 52-59 of the '869 Patent in view of the following references and grounds:

Grounds	Claim(s)	Basis
1	18, 20, 22, 24-32, 34, 36, 38-46, 48, 50, 52-59	§ 103 Ideses
2	18-20, 22, 24-34, 36, 38-48, 50, 52-59	§ 103 Ideses and Sharma

U.S. Patent Application Publication No. 2015/0213376 to Ideses ("Ideses," Ex. 1005), which application was filed on January 30, 2014, and published on July

30, 2015. Ideses is thus prior art under at least 35 U.S.C. § 102(a)(2).

U.S. Patent Application Publication No. 2016/0154960 to Sharma (“Sharma,” Ex. 1006), which application was filed on October 2, 2014, and published on June 2, 2016. Sharma is thus prior art under at least 35 U.S.C. § 102(a)(2).

Petitioner’s analysis also 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. *See Yeda Research v. Mylan Pharm. Inc.*, 906 F.3d 1031, 1041-1042 (Fed. Cir. 2018); 37 C.F.R. § 42.104(b); *see also K/S HIMPP v. Hear-Wear Techs., LLC*, 751 F.3d 1362, 1365-66 (Fed. Cir. 2014).

Each of these grounds is discussed in detail below.

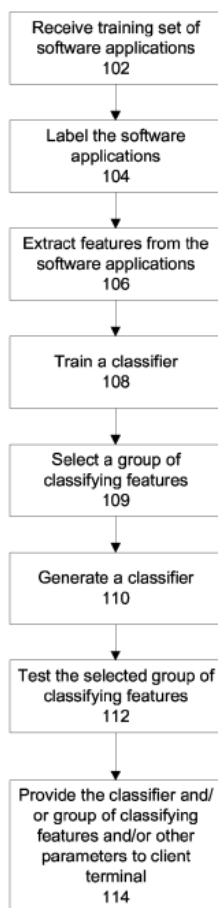
VIII. PRIOR ART OVERVIEW

A. Ideses

Ideses was not cited or referenced during the ‘869 patent prosecution. Ideses relates to “training a classifier” for classifying applications including, e.g., determining whether a software application is benign or whether it is harmful, malware, bothersome, adware, intrusive, spyware, or some other non-desirable classification type. Ex. 1005, Abstract, ¶¶0023, 0141. Ideses discloses devices, methods, and systems for training a classifier based on extracted features

(information) from software applications (e.g., “**static meta data** regarding the application, for example, icon, name, rating or other features” (emphasis added)). *Id.* at ¶¶0080-0089. Once the classifier is trained the model can analyze features from *new* software applications and predict the application’s classification type (e.g., “malware”). *Id.* at ¶¶0123-0124

Figure 1 illustrates an example of using data from multiple software applications as training data to train a classifier:

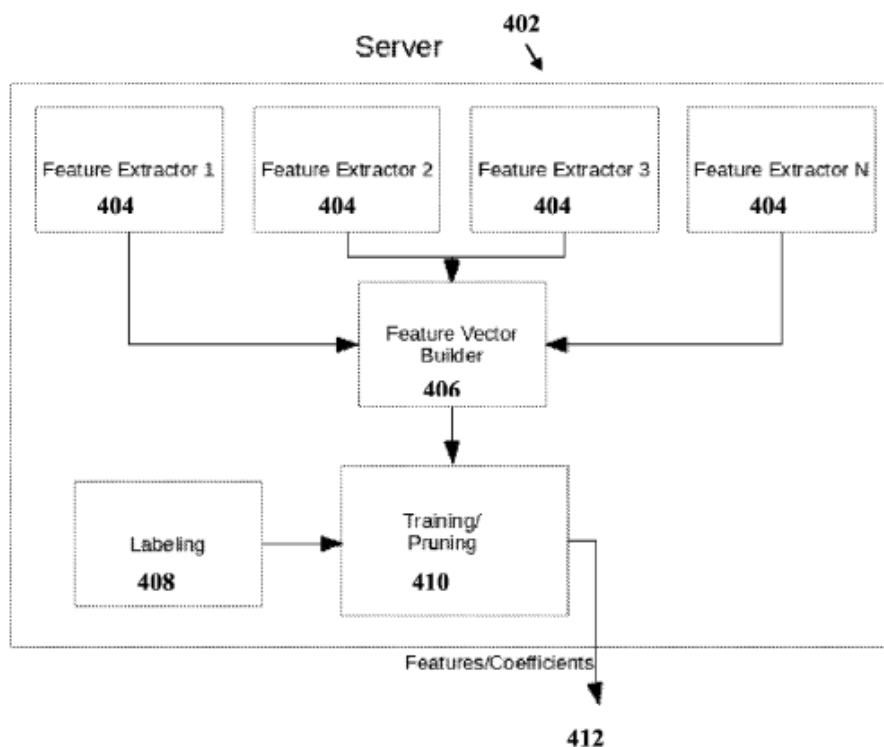


Ex. 1005, FIG. 1. At block 102, multiple software applications are received for training the classifier, which could be provided manually by the operator (e.g., the manufacturer) or automatically downloaded. *Id.* at ¶0080. At block 104, the multiple software programs are labeled with a classification type based on desired software classification type categories (e.g., previously labeled applications that were vetted by commercial companies). *Id.* at ¶¶0081- 0082. At block 106, features are extracted

from the software applications, (e.g., static data points). *Id.* at ¶¶0086, 0089. The features extracted may be all of the features or any combination of a sub-set of the features. *Id.* at ¶¶0086-0089. “[A]t block 108, a classifier is trained based on the set of extracted features (block 106) and labeling (block 104).” *Id.* at ¶0094. At block 110, a trained classifier is generated. *Id.* at ¶0115.

At step 109, it should be noted that a group of classifying features are selected for training the classifier. The group of selected features may be based on a number of considerations including a classification effectiveness rank and an identification of resource requirements for extracting said feature. Ex. 1005, ¶¶0098-0106. Ideses further discloses that the features not selected for the group for the classification process are pruned or not included. *Id.* at ¶0107.

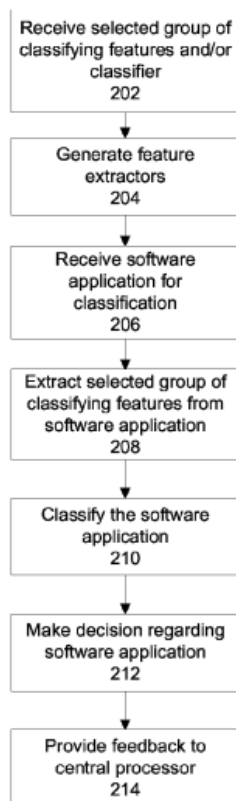
The feature selection and pruning process is further reflected in the process described in Figure 4:



Ex. 1005, FIG. 4.

Ideses discloses that a classifier trained by a training module 410 using a feature vector (extracted features) and associated label of a software application. A pruning module 410 reduces the size of the feature vector and/or the dimensionality of the feature vector. Pruning module 410 performs pruning so that the classifier maintains a certain level of accurate classification, while being able to perform the classification during run-time at a resource limited client terminal (e.g., mobile device). Ex. 1005, ¶0153.

Figure 2 depicts a “method of classification” using the generated classifier:



Ex. 1005, FIG. 2. At block 206, a software application is received by a mobile device (e.g., downloaded from the internet) or requested to be installed. *Id.* at ¶0131. At block 208, features are extracted from the software application (e.g., static meta data). *Id.* at ¶¶0089, 0132. At block 210, the software application block is classified (e.g., “as benign”). *Id.* at ¶0135. At block 212, a course of action is decided for the software application based on the classification (e.g., installation, “if the classification type is determined to be benign,” or deleted, “if the classification type

is determined to be harmful”) *Id.* at ¶0141. At block 214, the mobile device reports the classification type outcome to a server. *Id.* at ¶0144.

B. Sharma

Sharma relates to systems, methods, and computer readable medium for training a risk rating framework (RRF) system for assessing risk of a mobile application. Ex. 1006, Abstract, ¶0024. According to Sharma, the RRF system evaluates the risk and vulnerability of mobile computer applications using one or more classifiers. *Id.* at ¶0024. The one or more classifiers use manifest and static features computationally extracted from the mobile computer application and indicates the probability of being malware based on those features. *Id.*

Figure 1B illustrates “an overview of a RRF system”:

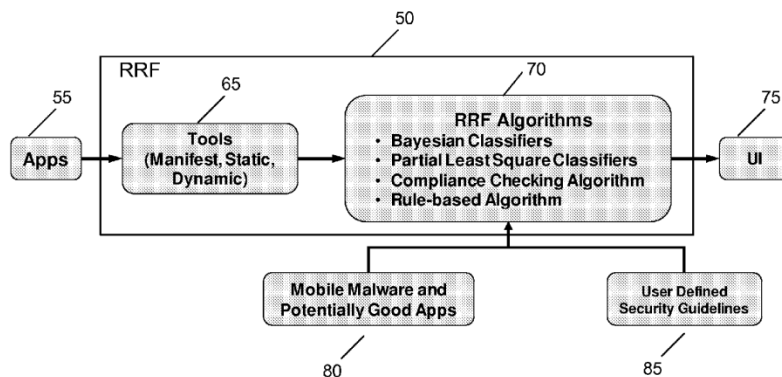


FIG. 1B

As shown in the figure, “classifiers 70 are used to analyze computer applications 55.” *Id.* at ¶0027. Tools 65 are programmed to extract certain features from computer applications 55 for analysis, which are used by classifiers 70 to evaluate risk. *Id.* Some of the analysis tools 65 include manifest, static, and dynamic analysis tools. *Id.* A manifest analysis tool may extract features and information from an application’s manifest file, which contains information regarding files that are included in the computer application package. *Id.* “A static analysis tool may extract features and information regarding an applications' software code, such as, code size, run time, and the like.” *Id.* “A dynamic analysis tool may extract features and information related to the computer application when it is running or operating, such as, incoming and outgoing application calls, and the like.” *Id.* According to Sharma, analysis of manifest, static, and dynamic features can help identify computer applications that exhibit malware or malicious characteristics. *Id.*

IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Ground 1: Claims 18, 20, 22, 24-32, 34, 36, 38-46, 48, 50, 52-59 are obvious under 35 U.S.C. § 103 by Ideses

1. Claim 18

[18.0] A computer-implemented method comprising:

To the extent the preamble is limiting, Ideses discloses “a *method* for training a classifier for classifying applications on a resource limited client terminal.” Ex.

1005, ¶0006; Ex. 1002, ¶93-94. Ideses discloses a system, a method, and/or a *computer* program product. “The *computer* program product may include a *computer* readable storage medium (or media) having *computer* readable program instructions thereon for causing a processor to carry out aspects of the present invention.” Ex. 1005, ¶0063. Since Ideses’ *method* is implemented on a *computer*, such as a client terminal, Ideses discloses a *computer-implemented method*.¹ Ex. 1002, ¶93.

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

Ideses discloses this limitation. Ex. 1002, ¶96.

First, Ideses discloses that “software applications are labeled with a **classification type** based on desired software application classification categories.” Ex. 1005, ¶0081. “For example, the classification types denote **desired or undesired software applications**” (*harmful or benign executable file*). *Id.*; Ex. 1002, ¶97-99.

Second, Ideses discloses that “**features are extracted** from the software applications” (*identifying*). Ex. 1005, ¶0086. “[E]xtracted features may be any feature that describes the software application. [T]he extracted features **may contain**

¹ **Emphasis** in quoted material has been added. Claim terms are presented in *italics*.

static meta data regarding the application, for example, icon, name, rating or other features.” (*static data points*). *Id.* at ¶0089; Ex. 1002, ¶¶57-58, 100-107.

Third, Ideses discloses that the extracted features such as static meta data are utilized in a group of classifying features for classification. For example, “a complete set of features is extracted from each application. Alternatively, individual or groups of features are extracted from each application, for example, as **features are being evaluated for inclusion in the group of classifying features**” (*indicative of either a harmful or benign executable file*). Ex. 1005, ¶0086; Ex. 1002, ¶108-112.

[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;

Ideses discloses this limitation. Ex. 1002, ¶113.

First, Ideses discloses categories of files including harmful and benign files. “[T]he multiple software applications are labeled with a classification type based on desired software application classification categories. For example, the **classification types denote desired or undesired software applications**” (*plurality of categories of files including harmful files and benign files*). Ex. 1005, ¶0081. In another example, Ideses teaches “**the classification type is benign or adware.**” *Id.*; Ex. 1002, ¶114-115.

Second, Ideses discloses that the extracted features (e.g., *static data points*) are *associated* with one of the classification types (*categories of files*) using a classifier. For example, “a complete set of features is extracted from each application. Alternatively, individual or groups of features are extracted from each application, for example, as **features are [sic] being evaluated for inclusion in the group of classifying features.**” Ex. 1005, ¶0086; Ex. 1002, ¶116-117.

“[T]he features are stored as feature vectors, which may be used for **training the classifier**” (*associating*). Ex. 1005, ¶0091; Ex. 1002, ¶118. Ideses further explains an association of extracted features (e.g., *static data points*) using classification labeling (e.g., *harmful files and benign files*), “**a classifier is trained based on the set of extracted features (block 106) and software classification labeling (block 104).**” Ex. 1005, ¶0094. “A single classifier or multiple classifiers may be used. For example, a combination of classifiers may be applied to classify feature vectors, for example a cascade of classifiers, a boosting topology of classifiers, or a parallel classification scheme.” *Id.*; Ex. 1002, ¶119-120.

[18.3] identifying an executable file to be evaluated;

Ideses discloses this limitation. Ex. 1002, ¶121.

First, Ideses discloses identifying an executable file as a particular “**software application**” (*executable file*) “**received** at the mobile device, for example, the software application is downloaded from the internet, **loaded** using physical computer readable media, **uploaded** by a third party, or other methods of **receipt**” (*identifying*). Ex. 1005, ¶0131; Ex. 1002, ¶122-124.

Second, Ideses discloses that the executable file is to be evaluated. “[T]he software application is **classified**” (*evaluated*) “by a run-time classification module 314.” Ex. 1005, ¶0135. “[T]he software application is **classified**, for example, as benign or adware” (*categories of files including harmful files and benign files*). *Id.*; Ex. 1002, ¶125-128.

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

Ideses discloses this limitation. Ex. 1002, ¶129.

First, Ideses discloses that “**features are extracted from the software applications**” (*from the executable file*). Ex. 1005, ¶0086. “[E]xtracted features may be any feature that describes the software application.” *Id.* at ¶0089. Ideses discloses that “features are **extracted based on the software application**, for example by the generated feature **extraction modules** (block 204).” *Id.* at ¶0132. “[T]he **features**

are *extracted* by the **generated feature extractors** (block 204). *Id.*; Ex. 1002, ¶¶59-62, 130-134.

Second, the ‘869 patent describes “[a] static data point may be a point of reference used to evaluate an executable file. As examples, 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.” Ex. 1001, 5:61-67. Other examples are described in TABLE 1.1:

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 heap	file OS	special build	Code Characteristics Anomaly
size of stack	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

Ex. 1001, TABLE 1.1

Ideses discloses such static data points. For example, “the ***extracted features*** may contain ***static meta data regarding the application***” (*static data points from the executable file*). Ex. 1005, ¶0089. “In another example, the extracted features may contain information regarding the executable code and/or software package, for example, in the form of byte code [(e.g. “package code,” “product code”)], resources [(e.g. “resources, “resource information”)], permissions [(e.g. “certificate information”)], or other features.” *Id.*; Cf. Ex. 1001, 5:61-67, TABLE 1.1; Ex. 1002, ¶¶54-56, 59-62, 135-141.

[18.5.a] generating a feature vector from the plurality of static data points

Ideses discloses this limitation. Ex. 1002, ¶142.

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.

[18.5.b] using a classifier trained to classify the static data points based on training data,

Ideses discloses this limitation. Ex. 1002, ¶145.

First, Ideses discloses that a classifier is trained. “[A]t 108, a **classifier is trained based on the set of extracted features** (block 106) and software classification labeling (block 104). Optionally, a learning module 306 (e.g., stored on memory 304) of a machine learning algorithm is applied to **train the classifier.**” Ex. 1005, ¶0094; Ex. 1002, ¶146-147. Ideses discloses “at 202, the group of classifying features, the **trained classifier**, and/or the pruned list of features (e.g., feature vector) is received by the mobile device.” Ex. 1005, ¶0124.

Second, Ideses discloses that the feature vector is generated [18.5.a] **using a trained classifier**. “[T]he **features extracted are based on the** group of classifying features and/or **trained classifier** received from the central server (block 202).” Ex. 1005, ¶0132; Ex. 1002, ¶¶60, 148-149. Ideses further discloses “optionally a learning module 306 (e.g., stored on memory 304) of a machine learning algorithm is applied to train the classifier” and references supervised and unsupervised learning modules for training the classifier. Ex. 1005, ¶¶0094, 0096; Ex. 1002, ¶153-157.

To the extent Patent Owner argues Ideses does not disclose *using a classifier trained to classify the static data points based on training data*, a POSITA would have found this limitation obvious. A POSITA would have understood that the feature vector is generated depending on the results of training the classifier *based on training data*. Ex. 1002, ¶¶62-63, 151-152. For example, Ideses discloses “[a]

classifier is trained by a training module 410.” Ex. 1005, ¶¶0153. A POSITA would have recognized that the features extracted based on the trained classifier necessarily use training data to train the classifier. Ex. 1002, ¶¶62-68, 149.

A POSITA would have further recognized that features extracted to generate the feature vector are based on the trained classifier *to classify the static data points*. Ex. 1002, ¶¶64-67, 150. For example, Ideses discloses “[t]he **group of classifying features** [(e.g., feature vector)] has been **selected based on the classification effectiveness** and/or based on resource requirements of each feature.” Ex. 1005, ¶¶0124, 0071; *See also id.* at ¶0132. A POSITA would have recognized that the feature vector is generated based on the described extraction and/or selection in Ideses. Ex. 1002, ¶¶62-63, 150-151.

Further, a POSITA would have recognized that training a classifier includes generating feature vectors to train the classifier. Ex. 1002, ¶¶67-71, 153-154. Ideses discloses “a machine learning algorithm is applied to train the classifier.” Ex. 1005 ¶0094; Ex. 1002, ¶155-158. Accordingly, a POSITA would have recognized that the machine learning algorithm needs training data using which it can train the classifier.

[18.5.c] the training data comprising files known to fit into one of the plurality of categories of files,

Ideses discloses this limitation. Ex. 1002, ¶159.

Ideses discloses, “at 102, **multiple software applications for *training* the classifier** are received, for example, by central processor” (*training data*). Ex. 1005, ¶0080. “The software applications may be manually provided by an operator of the system (e.g., the manufacturer), provided by software application manufacturers, automatically downloaded from the internet, provided by updates from mobile devices.” *Id.*; Ex. 1002, ¶¶62, 68-71, 160.

“[A]t 104 the multiple software applications are **labeled with a classification type** based on desired software application classification categories. For example, **the classification types denote desired or undesired software applications**” (*fit into one of the plurality of categories of files*). Ex. 1005, ¶0081. “Examples of labels include: Adware, Goodware, Intrusive Adware, or other classification types may be used.” Ex. 1005, ¶0083; Ex. 1002, ¶¶64-66, 163.

[18.5.d] wherein one or more features of the feature vector are selectively turned on or off

Ideses discloses this limitation. Ex. 1002, ¶¶164, 168.

Ideses discloses “**selecting** a set of classifying features extractable from software applications, in order to generate classifiers, provide the **selected group** of classifying features (e.g., **feature vectors**), select coefficients and/or other parameters for classification of software applications. Optionally, the **selected**

group of classification features is **a sub-set of the available features (all or some) that may be extracted** from a software application. The complete set of features may be pruned to generate the selected group.” Ex. 1005, ¶0044.

More particularly, “**features** (or their equivalents) that do not contribute (or do not significantly contribute) to the classification process **are pruned or not included**” (*selectively turned off*). Ex. 1005, ¶0107; Ex. 1002, ¶165. “Alternatively 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 a larger group” (*selectively turned on*). Ex. 1005, ¶0071; Ex. 1002, ¶166-167.

[18.5.e] 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

Ideses discloses this limitation. Ex. 1002, ¶169.

For example, Ideses discloses “features are selected **based on an identified classification effectiveness rank for each of the features**” (*evaluation of a value of one of the plurality of static data points*). Ex. 1005, ¶0045. “[C]lassification effectiveness sometimes refers to the contribution of the extracted feature to the accuracy of classification.” *Id.* “[I]f the accuracy of classification with the feature is 95%, and the accuracy of classification without the feature is 50%, the accuracy in classification based on the feature is 45%, which may [be] **significant for**

classification effectiveness” (*within a predetermined range*). *Id.* “Significant features, such as those that have a large effect on the classification outcomes, may be retained. Insignificant features, such as those that have a small, negligible or no effect on classification outcome may be pruned.” *Id.* at ¶0106; Ex. 1002, ¶169-172.

In another example, “the group of classifying features is **selected to reduce the dimensionality of the feature vector**” (*evaluation of a value of one of the plurality of static data points*). Ex. 1005, ¶0105. “[F]eatures that correspond to coefficients **having a value of zero** are pruned or not included” (*within a predetermined range*). *Id.* at ¶0107. Selection of the group of classifying features based on coefficients with zero value may be denoted by the expression:

$$\text{Feature}_i \in F \setminus f | (\text{Coefficient}_i) \neq 0$$

Id. at ¶0109.

Further, Ideses discloses that “**features with a low magnitude value of the coefficient are identified for pruning** and/or are not included” (*within a predetermined range*). *Id.* at ¶0107. Selection of the group of classifying features based on coefficients with low values may be denoted by the expression:

$$\text{Feature}_i \in F \setminus f | (\text{Coefficient}_i) \geq \alpha$$

Id. at ¶0110. Accordingly, Ideses discloses at least three examples of features (e.g., *static data points*) evaluated for inclusion (e.g., *selectively turned on or off*) based

on whether a value of one of the features is within a predetermined range. Ex. 1002, ¶173-175.

[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.

Ideses discloses this limitation. Ex. 1002, ¶176.

First, Ideses discloses that the “extracted features are collated into **feature vectors**. The **feature vectors** may be stored in buffers, which may provide for easy serial access.” Ex. 1005, ¶0134; *see* [18.5.a] - [18.5.e]; Ex. 1002, ¶¶64-67, 178.

Second, Ideses discloses evaluation of the feature vectors using a machine learning module. “[T]he software application is **classified based on the group of classifying features that have been computed by the feature extractor** (block 208).” Ex. 1005, ¶0137. Ideses discloses that “features that correspond to coefficients having a value of zero are pruned or not included, for example, **when classification is performed based on a support vector machine (SVM)**” (*evaluating using a machine learning module*). Ex. 1005, ¶0107; Ex. 1002, ¶¶177, 179.

Third, Ideses discloses “[a]t 210, **the software application is classified**, for example, by a **run-time classification module** 314 stored on memory 312.” Ex. 1005, ¶0135. “[T]he classification module labels the software application.” *Id.*

“[T]he software application is classified, for example, as benign or adware”
(whether the executable file fits into one of the categories of files). *Id.*; see also
[18.5.c]; Ex. 1002, ¶¶64-67, 180-181.

Ideses accordingly renders claim 18 obvious because it discloses each limitation as recited in the claim. One skilled in the art would have a reasonable expectation of successfully achieving the method of claim 18 because the disclosure of Ideses coupled with the knowledge of one skilled in the art involves nothing more than the use of known components in accordance with their conventional uses in a predictable art.² Ex. 1002, ¶¶54-56, 59-71, 151-158.

2. Claim 20

[20.0] The computer-implemented method according to claim 18,

See [18.0] - [18.6].

² The reasonable expectation of success described here is likewise applicable to the entirety of claim 18, independent claims 32 and 46, and all dependent claims listed in Ground 1.

[20.1] wherein the machine learning model comprises an artificial neural network.

Ideses discloses this limitation. Ex. 1002, ¶184.

First, Ideses discloses use of machine learning models generally, “the learning module performs the **machine learning** and/or classifier training.” Ex. 1005, ¶0095; *see also* [18.6]; Ex. 1002, ¶185.

Second, Ideses discloses specific “[e]xamples of software modules to train the classifier include: *Neural Networks*” (*artificial neural network*). Ex. 1005, ¶0096; Ex. 1002, ¶186-187.

3. Claim 22

[22.0] The computer-implemented method according to claim 18,

See [18.0] - [18.6].

[22.1] wherein the machine learning model comprises a machine learning decision tree.

Ideses discloses this limitation. Ex. 1002, ¶189.

First, Ideses discloses use of machine learning models generally, “the learning module performs the **machine learning** and/or classifier training.” Ex. 1005, ¶0095; *See also* [18.6]; Ex. 1002, ¶190.

Second, Ideses discloses specific “[e]xamples of software modules to train the classifier include: . . . **Decision Trees**” (*machine learning decision tree*). Ex. 1005, ¶0096; Ex. 1002, ¶191-196.

4. Claim 24

[24.0] The computer-implemented method according to claim 18,

See [18.0] - [18.6].

[24.1] wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.

Ideses discloses this limitation. Ex. 1002, ¶198.

First, Ideses discloses “extracted features are collated into **feature vectors**. The **feature vectors** may be stored in buffers, which may provide for easy serial access” and “the software application is **classified based on the group of classifying features that have been computed by the feature extractor** (block 208)” (*evaluating the feature vector*). Ex. 1005, ¶¶0134, 0137; Ex. 1002, ¶199; See also [18.6].

Second, Ideses discloses, “features that correspond to coefficients having a value of zero are pruned or not included, for example, **when classification is**

performed based on a *support vector machine (SVM)*” (*evaluating the feature vector using support vector processing*). Ex. 1005, ¶0107; Ex. 1002, ¶200-201.

5. Claim 25

[25.0] The computer-implemented method according to claim 18,

See [18.0] - [18.6].

[25.1] wherein the machine learning model comprises a clustering model.

Ideses discloses this limitation. Ex. 1002, ¶203.

First, Ideses discloses use of machine learning models generally, “the learning module performs the **machine learning** and/or classifier training.” Ex. 1005, ¶0095;
See also [18.6].

Second, Ideses discloses specific “non-supervised approach is used in which labeling is based on **clusterization**. Optionally, clusters are generated automatically using a non-supervised and/or a semi-supervised **clustering** software module” (*clustering model*). Ex. 1005, ¶0085; Ex. 1002, ¶204-205.

6. Claim 26

[26.0] The computer-implemented method according to claim 18,

See [18.0] - [18.6].

[26.1] wherein the executable file is identified in response to a detected condition.

Ideses discloses this limitation. Ex. 1002, ¶207.

Ideses discloses a number of *detected conditions* in which an *executable file is identified*. For example, “at 206, a software application is **received at the mobile device**, for example, the software application is **downloaded from the internet, loaded using physical computer readable media, uploaded by a third party, or other methods of receipt**. Optionally, the software application is **requesting** (automatically or manually) **to be installed on the mobile device**.” Ex. 1005, ¶0131; Ex. 1002, ¶208-210.

7. Claim 27

[27.0] The computer-implemented method according to claim 26,

See [26.0] - [26.1].

[27.1] wherein the detected condition is user request for a file download.

Ideses discloses this limitation. Ex. 1002, ¶212.

Ideses discloses “at 206, a **software application is received at the mobile device**, for example, the **software application is downloaded** from the internet, loaded using physical computer readable media, uploaded by a third party, or other

methods of receipt” (*user request for a file download*). Ex. 1005, ¶0131; Ex. 1002, ¶213-214.

8. Claim 28

[28.0] The computer-implemented method according to claim 26,

See [26.0] - [26.1].

[28.1] wherein the detected condition is the detection of a new file attempting to execute.

Ideses discloses this limitation. Ex. 1002, ¶216.

Ideses discloses “[t]he method of FIG. 2 may be automatically executed by software, for example, the mobile device automatically updates itself with the latest feature set, **automatically detects software trying to install itself**, automatically classifies the software, and/or automatically removes possibly harmful software” (*detection of a new file attempting to execute*). Ex. 1005, ¶0148; Ex. 1002, ¶217.

9. Claim 29

[29.0] The computer-implemented method according to claim 18,

See [18.0] - [18.6].

[29.1] wherein the plurality of static data points represent predefined character strings in the executable file.

Ideses discloses this limitation. Ex. 1002, ¶220.

Ideses discloses that “[t]he extracted features may be **any feature that describes the software application.**” Ex. 1005, ¶0089. “For example, the extracted features may contain static meta data regarding the application, for example, **icon, name,** rating or other features.” *Id.*; Ex. 1002, ¶¶221-222.

A POSITA would have understood that “any feature that describes the software application,” including the specific examples of “icon” and “name,” which are commonly held in data structure like character strings, would include *predefined character strings in the executable file*. *Id.*; Ex. 1002, ¶¶47-48, 223-224.

10. Claim 30

[30.0] The computer-implemented method according to claim 18,

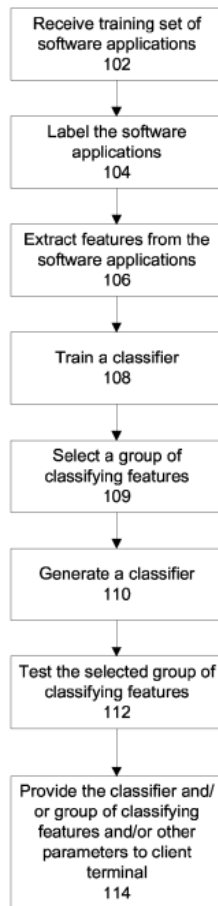
See [18.0] - [18.6].

[30.1] wherein a determination of whether the executable file is harmful is used to retrain the classifier.

Ideses discloses this limitation. Ex. 1002, ¶226.

First, Ideses discloses “at 214, the **mobile device reports back to the server.**” Ex. 1005, ¶0144. “[T]he **classification type outcome** is reported” (*determination of whether the executable file is harmful*). *Id.*; Ex. 1002, ¶¶227-228.

Second, Ideses discloses “information regarding the software application is sent back as part of a **feedback loop.**” Ex. 1005, ¶0144. And further, “the central server **learns about the existence of new software applications based on the feedback.**” *Id.* at ¶0145. “Referring back to FIG. 1, optionally, at 102, **multiple software applications for training the classifier are received**, for example, by central processor 302.” *Id.* at ¶0080.



A POSITA would have understood that the central server learns about the existence of new software applications and their respective classification types as part of the feedback loop in order to *retrain the classifier*. Ex. 1002, ¶229-230.

11. Claim 31

[31.0] The computer-implemented method according to claim 18,

See [18.0] - [18.6].

[31.1] further comprising preventing execution of the executable file based on classifying the executable file as harmful.

Ideses discloses this limitation. Ex. 1002, ¶233.

Ideses discloses “at 212, a **course of action** is decided for the software application **based on the automated classification.**” Ex. 1005, ¶0141. “[T]he software application is not installed, deleted, flagged, or otherwise *prevented from functioning* on the mobile device, for example, if the **classification type is determined to be *harmful***, malware, bothersome, adware, intrusive, spyware, or other non-desirable classification types.” *Id.*; Ex. 1002, ¶¶234-235.

12. Claim 32

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

To the extent the preamble is limiting, Ideses discloses “[t]he present invention may be a system, a **method,**” (*process*) “and/or a computer program product. The computer program product may include a **computer readable storage medium** (or media) **having computer readable program instructions thereon.**” Ex. 1005, ¶0063. “The **computer readable storage medium can be a tangible device** that can retain and store **instructions for use by an instruction execution device**” (*cause a processor to execute*). Ex. 1005, ¶0064; Ex. 1002, ¶¶237-238.

Further, Ideses discloses a specific embodiment of a processor executing instructions as data modules, “[s]ystem 300 includes a central **processor** 302 with a memory 304 for storing **data modules** 306 thereon.” Ex. 1005, ¶0073. Examples of execution of a process include, “[c]entral processor 302 may have sufficient **processing ability to train a classifier** based on a full set of features, for example, extracted from software applications.” *Id.* **Modules 306 may include a feature extraction and/or pruning system.** *Id.* at ¶0074. **“Modules 314 may include a run-time feature extractor** for extracting features from a software application for classification based on the selected group of classifying features.” *Id.* at ¶0076; Ex. 1002, ¶239-240.

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

See [18.1]. Ex. 1002, ¶241.

[32.2] 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;

See [18.2]. Ex. 1002, ¶242.

[32.3] identifying an executable file to be evaluated;

See [18.3]. Ex. 1002, ¶243.

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

See [18.4]. Ex. 1002, ¶244.

[32.5.a] generating a feature vector from the plurality of static data points

See [18.5.a]. Ex. 1002, ¶245.

[32.5.b] using a classifier trained to classify the static data points based on training data,

See [18.5.b]. Ex. 1002, ¶246.

[32.5.c] the training data comprising files known to fit into one of the plurality of categories of files,

See [18.5.c]. Ex. 1002, ¶247.

[32.5.d] wherein one or more features of the feature vector are selectively turned on or off

See [18.5.d]. Ex. 1002, ¶248.

[32.5.e] 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

See [18.5.e]. Ex. 1002, ¶249.

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

See [18.6]. Ex. 1002, ¶250.

13. Claim 34

[34.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[34.1] wherein the machine learning model comprises an artificial neural network.

See [20.1]. Ex. 1002, ¶252.

14. Claim 36

[36.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[36.1] wherein the machine learning model comprises a machine learning decision tree.

See [22.1]. Ex. 1002, ¶254.

15. Claim 38

[38.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[38.1] wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.

See [24.1]. Ex. 1002, ¶256.

16. Claim 39

[39.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[39.1] wherein the machine learning model comprises a clustering model.

See [25.1]. Ex. 1002, ¶258.

17. Claim 40

[40.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[40.1] wherein the executable file is identified in response to a detected condition.

See [26.1]. Ex. 1002, ¶260.

18. Claim 41

[41.0] The computer-readable storage device according to claim 40,

See [40.0] - [40.1].

[41.1] wherein the detected condition is user request for a file download.

See [27.1]. Ex. 1002, ¶262.

19. Claim 42

[42.0] The computer-readable storage device according to claim 40,

See [40.0] - [40.1].

[42.1] wherein the detected condition is the detection of a new file attempting to execute.

See [28.1]. Ex. 1002, ¶264.

20. Claim 43

[43.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[43.1] wherein the plurality of static data points represent predefined character strings in the executable file.

See [29.1]. Ex. 1002, ¶266.

21. Claim 44

[44.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[44.1] wherein a determination of whether the executable file is harmful is used to retrain the classifier.

See [30.1]. Ex. 1002, ¶268.

22. Claim 45

[45.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[45.1] wherein the process further comprises preventing execution of the executable file based on classifying the executable file as harmful.

See [31.1]. Ex. 1002, ¶270.

23. Claim 46

[46.0] A system comprising:

To the extent the preamble is limiting, Ideses discloses a “*/s/system* 300 includes a central processor 302 with a memory 304 for storing data modules 306 thereon.” Ex. 1005, ¶0073; Ex. 1002, ¶271-272.

[46.1] at least one memory;

Ideses discloses this limitation. Ex. 1002, ¶273.

Ideses discloses a “[s]ystem 300 includes a central processor 302 with a *memory* 304 for storing data modules 306 thereon.” Ex. 1005, ¶0073; Ex. 1002, ¶274.

[46.2] at least one processor configured to perform operation of:

Ideses discloses this limitation. Ex. 1002, ¶275.

Ideses discloses a “[s]ystem 300 includes a **central processor** 302 with a memory 304 for storing data modules 306 thereon.” Ex. 1005, ¶0073. Ideses further discloses that “computer readable program instructions may be provided to a **processor.**” *Id.* at ¶0068. And, the **instructions “execute via the processor”** (*configured to perform operation*). *Id.*; Ex. 1002, ¶276-277.

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

See [18.1]. Ex. 1002, ¶278.

[46.4] 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; and

See [18.2]; Ex. 1002, ¶279.

[46.5] at least one processor configured to perform operation of:

See [46.2]; Ex. 1002, ¶280.

[46.6] identifying an executable file to be evaluated;

See [18.3]; Ex. 1002, ¶281.

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

See [18.4]; Ex. 1002, ¶282.

[46.8.a] generating a feature vector from the plurality of static data points

See [18.5.a]; Ex. 1002, ¶283.

[46.8.b] using a classifier trained to classify the static data points based on training data,

See [18.5.b]; Ex. 1002, ¶284.

[46.8.c] the training data comprising files known to fit into one of the plurality of categories of files,

See [18.5.c]; Ex. 1002, ¶285.

[46.8.d] wherein one or more features of the feature vector are selectively turned on or off

See [18.5.d]; Ex. 1002, ¶286.

[46.8.e] 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

See [18.5.e]; Ex. 1002, ¶287.

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

See [18.6]; Ex. 1002, ¶288.

24. Claim 48

[48.0] The system according to claim 46,

See [46.0] - [46.9].

[48.1] wherein the machine learning model comprises an artificial neural network.

See [20.1]; Ex. 1002, ¶290.

25. Claim 50

[50.0] The system according to claim 46,

See [46.0] - [46.9].

[50.1] wherein the machine learning model comprises a machine learning decision tree.

See [22.1]; Ex. 1002, ¶292.

26. Claim 52

[52.0] The system according to claim 46,

See [46.0] - [46.9].

[52.1] wherein evaluating the feature vector using the machine learning model comprises evaluating the feature vector using support vector processing.

See [24.1]; Ex. 1002, ¶294.

27. Claim 53

[53.0] The system according to claim 46,

See [46.0] - [46.9].

[53.1] wherein the machine learning model comprises a clustering model.

See [25.1]; Ex. 1002, ¶296.

28. Claim 54

[54.0] The system according to claim 46,

See [46.0] - [46.9].

[54.1] wherein the executable file is identified in response to a detected condition.

See [26.1]; Ex. 1002, ¶298.

29. Claim 55

[55.0] The system according to claim 54,

See [54.0] - [54.1].

[55.1] wherein the detected condition is user request for a file download.

See [27.1]; Ex. 1002, ¶300.

30. Claim 56

[56.0] The system according to claim 54,

See [54.0]- [54.1].

[56.1] wherein the detected condition is the detection of a new file attempting to execute.

See [28.1]; Ex. 1002, ¶302.

31. Claim 57

[57.0] The system according to claim 46,

See [46.0] - [46.9].

[57.1] wherein the plurality of static data points represent predefined character strings in the executable file.

See [29.1]; Ex. 1002, ¶304.

32. Claim 58

[58.0] The system according to claim 46,

See [46.0] - [46.9].

[58.1] wherein a determination of whether the executable file is harmful is used to retrain the classifier.

See [30.1]; Ex. 1002, ¶306.

33. Claim 59

[59.0] The system according to claim 46, further comprising at least one processor configured to perform operation of:

See [46.0] - [46.9].

[59.1] preventing execution of the executable file based on classifying the executable file as harmful.

See [31.1]; Ex. 1002, ¶308.

B. Ground 2: Claims 18-20, 22, 24-34, 36, 38-48, 50, 52-59 are obvious under 35 U.S.C. § 103 over Ideses in view of Sharma

1. Reasons to Combine Ideses and Sharma

A POSITA would have been motivated to combine the teachings of Ideses and Sharma for multiple reasons, including to produce the obvious, beneficial, and predictable results of detecting and preventing malware. Doing so would have been within the POSITA's skillset and would produce beneficial, predictable results. Ex. 1002, ¶309.

When considering the teachings of Ideses, a POSITA would also have considered the teachings of Sharma. Both Ideses and Sharma are analogous prior art in the same field of endeavor as the '869 patent – detecting and preventing malware using learning classifiers. Ex. 1005, Abstract, ¶¶0023, 0058; Ex. 1006, Abstract, ¶0024-0026; Ex. 1002, ¶310. In particular, both Ideses and Sharma describe static analysis tools.

A POSITA would have been specifically motivated to incorporate Sharma's extraction techniques to expand Ideses' ability to utilize extracted parameters for training and applying a classifier. Ex. 1002, ¶311. The combination of Ideses and Sharma would have simply combined prior art elements (techniques of feature

extraction, including in a non-executing modality in Ideses; static and manifest extraction tools in Sharma) according to known devices (systems of training and applying a classifier) to yield predictable results (better-trained systems for training and applying a classifier) as Ideses intended, and thus would have been obvious to a POSITA. Ex. 1002, ¶312.

First, a POSITA would have been specifically motivated to incorporate Sharma's known techniques for feature extraction because doing so would address a known inefficiency in Ideses. Ex. 1002, ¶313. In particular, Ideses recognizes that the method and cost of extracting features is an important consideration when determining which features to select, and that "the classification features are selected based on an identified resource requirements for extracting each of the features." Ex. 1005, ¶0046. However, Ideses fails to provide specific methods for feature extraction or an associated resource cost. Sharma specifically provides example tools and further describes cost considerations. Ex. 1006, ¶40-45; Ex. 1002, ¶314-317.

Second, a POSITA would have been specifically motivated to incorporate Sharma's disclosed features in training Ideses' machine learning model because doing so enhances Ideses' predictive performance. Ex. 1002, ¶318. It was well known that new, different, and/or additional features enhance the predictive quality of a machine learning classifier. Ex. 1002, ¶322-324. Ideses recognizes that "[t]he

extracted features may be any feature that describes the software application” in light of anti-malware solutions. Ex. 1005, ¶¶0089, 0005. While Ideses discloses a certain set of features, a POSITA would have been motivated to seek out other features (such as those disclosed in Sharma) in order to improve the feature selection taught by Ideses and accordingly enhance the predictive quality of Ideses’ machine learning model for malware. Ex. 1002, ¶¶320-321. In particular, Sharma discloses features not considered in Ideses that are specifically indicative of malware, “[t]he extracted features are used by the classifiers 70 to evaluate the risk of computer applications 55.” Ex. 1006, ¶0027. A POSITA would have been specifically motivated to incorporate Sharma’s additional features that are indicative of malware into Ideses in order to more effectively evaluate a software application beyond the features of Ideses, as motivated by Ideses. Ex. 1005, ¶0089; Ex. 1002, ¶322-323.

Third, a POSITA would have been specifically motivated to incorporate Sharma’s rule-based compliance checking and incorporation of a threat model to assist in identifying harmful and benign applications because doing so would improve Ideses’ predictive performance, improve Ideses’ efficiency, and expand the predictive ability of Ideses. Ex. 1002, ¶325.

As mentioned above, Ideses recognizes the difficulty in extracting features from an application, “the classification features are selected based on an identified

resource requirements for extracting each of the features. For example, the memory required to extract the feature, the processor usage required to extract the feature, the time to extract the feature, and/or other factors.” Ex. 1005, ¶0046. A POSITA would have specifically sought Sharma’s rule-based compliance checking and specific increase in Ideses’ system’s predictive functionality and run-time efficiency for applications with low quantities of extractable features and for applications that would require significant resource requirements to extract the features, in view of the resource requirements described in Ideses. Ex. 1002, ¶¶326-329. A POSITA would have also been motivated to incorporate rules-based compliance checking algorithms for additional feature generation for the reasons discussed above with respect to improved predictive performance. Ex. 1002, ¶330-332.

Utilizing Sharma’s techniques in Ideses’ system would have not only been advantageous, but a POSITA would have had a reasonable expectation of success in producing the method recited in claim 18 through this combination. Ex. 1002, ¶¶317, 324, 333. Such expectation is evident because the combination involved the use of

known techniques and components in accordance with their conventional uses in a predictable art.³ Ex. 1005, ¶0089; Ex. 1002, ¶¶54-56, 59-71, 317, 324, 333-334.

Petitioner incorporates by reference the analysis of the Ideses grounds above, which grounds will not be restated here. See Section IX.A., supra.

2. Elements [18.5.b], [32.5.b] and [46.5.b] (as recited above in Section IX.A.)

Ideses alone or in combination with Sharma discloses this limitation. Ex. 1002, ¶336.

To the extent Patent Owner argues Ideses does not disclose generating a feature vector *using a classifier trained to classify the static data points based on training data*, Sharma explicitly discloses *using a classifier trained to classify static data points based on training data*. Sharma discloses a training a risk rating framework (RRF) system for assessing risk of a mobile application, and in particular, a computer security system that evaluates the risk and vulnerability of mobile computer applications using one or more classifiers. Ex.1006, ¶0024.

³ The reasonable expectation of success described here is likewise applicable to the entirety of claim 18, independent claims 32 and 46, and all dependent claims listed in Ground 2 including those incorporated by reference.

First, Sharma discloses that “**a feature vector is constructed using a Bayesian model classifier.**” Ex.1006, ¶0047. Accordingly, Sharma explicitly discloses generating a feature vector *using a classifier*. Ex. 1002, ¶339.

Second, Sharma discloses that “[t]he classifiers are first trained and tested using an input data set that includes known-malware and known-benign computer applications.” Ex. 1006, ¶0025. Accordingly, Sharma explicitly discloses a classifier *trained based on training data*. Ex. 1002, ¶338.

Third, Sharma continues that “[f]or the Bayesian classifier, all the features are inputted to the classifier. The classifier analyses each feature and determines if each feature contributes to the probability that the computer application is malware. **The classifier is designed to consider each feature to contribute independently to the probability of being malware.**” Ex. 1006, ¶0047. Accordingly, Sharma explicitly discloses consideration by a classifier of the independent contribution of each feature (including features such as static data points, as described in Ideses and Sharma) to a probability of being malware, in other words, the *classifier trained to classify the static data points*. Ex. 1002, ¶340. Sharma therefore discloses generating a feature vector using a classifier trained to classify the static data points based on training data.

The combination of Ideses and Sharma accordingly renders claim 18 obvious because it discloses each limitation as recited in the claim. One skilled in the art would have a reasonable expectation of successfully achieving the method of claim 18 because the disclosure of Ideses coupled with Sharma involves nothing more than the use of known components in accordance with their conventional uses in a predictable art. Ex. 1002, ¶¶54-56, 59-71, 341.

3. Claim 19

[19.0] The computer-implemented method according to claim 18,

See [18.0] - [18.6].

[19.1] wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

Ideses alone or combined with Sharma renders this limitation obvious. Ex. 1002, ¶343.

First, Ideses explains that “extracted features may be varied, containing information from different modalities.” Ex. 1005, ¶0089. Ideses specifically contrasts run-time feature extraction, which necessarily requires decrypting or unpacking the executable file, with off-line feature extraction. “The **feature extraction may take place off-line, not part of a run-time operation.**” Ex. 1005, ¶0090; Ex. 1002, ¶345-348.

Ideses further explains that “[e]xtraction of the full set of features may not be possible during run-time on the mobile device. Extraction of the full set of features may take place at the central processor, **independently of run-time operation** of the mobile device.” Ex. 1005, ¶0092.

Ideses further provides a number of examples of static data points that are extracted without decrypting or unpacking the executable file. Ex. 1002, ¶349. “For example, the extracted features may contain static meta data regarding the application, for example, **icon, name, rating** or other features. In another example, the extracted features may contain information regarding the executable code and/or software package.” Ex. 1005, ¶0090; Ex. 1002, ¶¶349-350.

Second, to the extent Ideses does not explicitly disclose static data point extraction *without decrypting or unpacking the executable file*, Sharma renders this limitation obvious. Sharma specifically discloses, referring to FIG. 1B, “tools 65 are programmed to **extract certain features from the computer applications 55** for analysis. The extracted features are used by the classifiers 70 to evaluate the risk of computer applications 55.” Ex. 1006, ¶0027; Ex. 1002, ¶351-353.

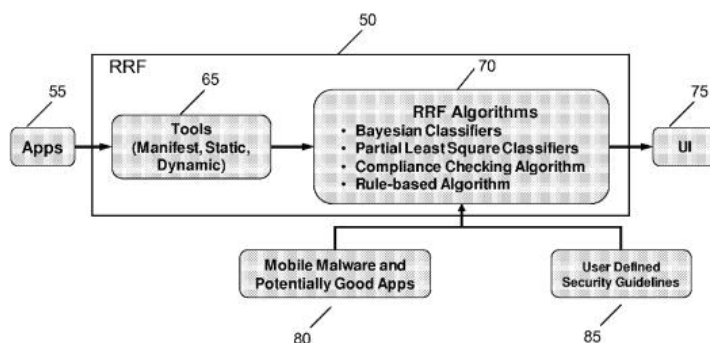


FIG. 1B

Sharma discloses specific extraction from manifest, static, and dynamic tools. For example, a “static analysis tool may extract features and information regarding an applications' software code, such as, code size, run time, and the like.” Ex. 1006, ¶0027.

Sharma specifically contrasts its static extraction with its dynamic analysis tool that “extract[s] features and information related to the computer application when it is running or operating,” which necessarily requires decrypting or unpacking the executable file. Ex. 1006, ¶0027; Ex. 1002, ¶354-355.

It would have been obvious to a POSITA to combine Ideses and Sharma to extract static data points without decrypting or unpacking the executable file. Ideses already discloses that “extracted features may be any feature that describes the software application. The extracted features may be varied, containing information

from different modalities.” Ex. 1005, ¶¶0089. The definition of features, as provided by the static analysis tool, in Sharma improves the prediction performance of Ideses. Ex. 1002, ¶¶309-339, 356-357.

A POSITA would have had a reasonable expectation of success in producing the method recited in claim 19 through the combination of Ideses with Sharma. In particular, the combination is nothing more than applying Ideses’ known techniques of feature extraction, including in a non-executing modality, to Sharma’s known static extraction tools to yield the predictable results of a system for training and applying a classifier based on “any feature that describes the software application” as Ideses intended, and thus would have been obvious to a POSITA. Ex. 1005, ¶¶0089, 0090; Ex. 1006, ¶0025; Ex. 1002, ¶¶317, 358.

4. Claim 33

[33.0] The computer-readable storage device according to claim 32,

See [32.0] - [32.6].

[33.1] wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

See [19.1]; Ex. 1002, ¶360.

5. Claim 47

[47.0] The system according to claim 46,

See [46.0] - [46.9].

[47.1] wherein the plurality of static data points are extracted without decrypting or unpacking the executable file.

See [19.1]; Ex. 1002, ¶362.

X. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE

A. THE BOARD SHOULD NOT DENY INSTITUTION UNDER SECTION 314(a)

The six factors considered for § 314 denial strongly favor institution. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (Mar. 20, 2020) (precedential).

1. Factor 1: Whether the district court granted a stay or a stay may be granted if a proceeding is instituted

No motions to stay have been entered or are currently pending in the co-pending parallel litigation. At a minimum, this factor is neutral because the Board should not presume how the district court would rule should a motion to stay be filed. *Sand Revolution II, LLC v. Continental Intermodal Group-Trucking LLC*,

IPR2019-01393, Paper 24, 7 (PTAB Jun. 16, 2020); *see also Dish Network L.L.C. v. Broadband iTV, Inc.*, IPR2020-01359, Paper 15 at 11 (Feb. 12, 2021).

2. Factor 2: Proximity of the court’s trial date to the Board’s projected statutory deadline for a final written decision

Trial in the co-pending parallel litigation is currently set for August 19, 2024, approximately 14 months from the filing date of this petition. The co-pending parallel litigation includes more than ten patents asserted by the Patent Owner, numerous counter claim patents asserted by the Defendants, four additional defendants, and three scheduled *Markman* hearings, with the Markman hearing for the ’869 patent docketed for July 25, 2023.

The five lawsuits are consolidated only for pre-trial purposes and each defendant will receive its own trial. Thus, the current trial date likely applies only to the first defendant to go to trial. Of the five defendants, Petitioner was served fourth. Accordingly, if the cases are tried in the order in which the lawsuits were filed, Petitioner’s trial date will likely be weeks or months after the August 2024 date.

Further, “additional supporting factors” are at play weighing towards a longer than average filing-to-trial window. First, the Kaspersky District Court litigation involves fourteen patents, including ten asserted by Patent Owner against Petitioner and four counterclaim infringement allegations brought by Petitioner. There are five

additional patents asserted by Patent Owner against the other defendants. Likewise, the non-Kaspersky defendants have asserted more than twenty additional counterclaim patents against Patent Owner. To accommodate the significant number of allegations, the Court has scheduled three Markman hearings. Accordingly, with multiple Markman hearings, a multitude of claims, and an inability to hold five trials at the same time, the “actual trial date” is more likely to be delayed past the scheduled August 19, 2024, trial date and after FWD here.

Even if the trial date remains as currently scheduled, there is likely sufficient time to complete the IPR proceeding prior to the trial date. Accordingly, this factor weighs heavily in favor of institution. *See Dish Network L.L.C.*, IPR2020-01359 at 16 (acknowledging that for a trial date scheduled nine months in the future much can change during that time). But even if the Board believes this factor weighs in favor of denial, the Board should balance all six factors when determining whether to apply discretionary denial. *Dish Network L.L.C.*, IPR2020-01359 at 14-15.

3. Factor 3: Investment in the parallel proceeding by the court and the parties

As indicated above, the parallel proceeding is in its early stages. The Markman hearing for claim construction of the ‘869 patent has not occurred yet and is scheduled for July 25, 2023. And fact discovery is not scheduled to close until

January 18, 2024. The deadline to add parties is not until October 3, 2023. Opening expert reports are not due until January 30, 2024, and expert discovery does not close until March 12, 2024. Therefore, the parties have invested minimal time in the parallel proceeding. The lack of investment weighs in favor of institution or is at least neutral.

4. Factor 4: Overlap between issues raised in the petition and in the parallel proceeding

Petitioner has raised Ideses and Sharma in preliminary invalidity contentions filed in the parallel litigation. However, updated infringement and invalidity contentions are not due until July 30, 2023. Therefore, the extent of overlap is unknown at this time. Further, the Petitioner reserves the right to stipulate to elimination of the overlap between this proceeding and the parallel litigation should the IPR be instituted. This would mitigate the concerns over duplication. *Sotera Wireless, Inc. v. Masimo Corp.*, IPR2020-01019, Paper 12, at 18-19 (PTAB Dec. 1, 2020); *SandRevolution II, LLC*, IPR2019-01393 at 11-12. Therefore, this factor weighs in favor of institution or is at least neutral.

5. Factor 5: Whether the petitioner and the defendant in the parallel proceeding are the same party

Petitioner and Webroot are parties in the parallel litigation. The Patent Owner is the petitioner in the parallel proceeding and the Petitioner is one of four defendants in the consolidated co-pending litigation. This factor is therefore neutral regarding institution. *See HP Inc. v. Slingshot Printing LLC*, IPR2020-01084, Paper 13 at 9 (Jan. 14, 2021); *see also Cisco Sys., Inc. v. Ramot at Tel Aviv Univ. Ltd.*, IPR2020-00122, Paper 15, at 11 (J. Crumbley dissenting) (PTAB May 15, 2020) (noting that disfavoring a defendant in parallel litigation is “contrary to the goal of providing district court litigants an alternative venue to resolve questions of validity.”).

6. Factor 6: Other circumstances that impact the Board’s exercise of discretion, including the merits

This factor includes consideration of the merits of the proceeding, which weighs in favor of institution. Neither Ideses nor Sharma were considered in prosecution. Since the art and arguments being asserted in this petition are materially different from those presented in prosecution, the considerations as to art previously before the USPTO, are favorable to institution. *Hewlett Packard Enterprise Co., et al. v. Billjco LLC*, IPR2022-00420, Paper 17, 6 (PTAB July 12, 2022).

Further, the Patent Owner has asserted the ‘869 patent against multiple defendants. Therefore, there is strong public interest in the Board instituting the IPR and adjudicating the validity of the patent. As such, this factor weighs in favor of institution.

B. THE BOARD SHOULD NOT DENY INSTITUTION UNDER GENERAL PLASTIC

The Board considers the *General Plastic* factors when multiple petitions are filed against the same patent. *See Gen. Plastic Indus. Co. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19, 16 (Sept. 6, 2016) (Section II.B.4.i. precedential). The ‘869 patent has been challenged in the Sophos PGR. Discretionary denial of this petition would unfairly deprive Kaspersky of access to *inter partes* review with respect to the ‘869 patent.

1. Factor 1: Whether the Same Petitioner Previously Filed a Petition

This factor weighs against denial because Petitioner Kaspersky is a *different* petitioner than the petitioner in the Sophos PGR. Moreover, there is not a significant relationship between Petitioner and Sophos. Although the Board held in *Valve Corp. v. Elec. Scripting Prods., Inc.*, IPR2019-00062, Paper 11 (Apr. 2, 2019) (precedential) that the application of the first *Generic Plastic* factor is not limited to

instances where multiple petitions are filed by the same petitioner, the facts here are distinguishable from *Valve*. First, in *Valve*, both the petitioner (*Valve*) and HTC (who filed the earlier IPR) were co-defendants in the same district court case and were accused of infringing the patent-at-issue based on the same product. *See Valve*, IPR2019-00062, at 9. However, in the present case, Petitioner was sued separately, accused of infringement based on different products, and is entitled to its own trial relating to its own products.

The Board routinely finds that “*General Plastic* and *Valve* do not apply” to the circumstances here, where the petitioners were sued independently, were sued on different products, and have no significant relationship. *See, e.g., NetNut Ltd. V. Bright Data Ltd.*, IPR2021-00465, Paper 11, at 8-11 (Aug. 12, 2021) (“*NetNut*”) (“Under these circumstances, we decline to extend *General Plastic* and *Valve* to the situation here: where a follow-on petition is filed by a petitioner who is not the same as previous petitioners and does not have any relationship, much less a significant relationship, with previous petitioners.”); *Sony Mobile Commc’ns AB. V. Ancora Techs., Inc.*, IPR2021-00663, Paper 17, 7-11 (Jun. 10, 2021) (“*Sony Mobile*”); *Shenzhen AOTO Elecs. Co., Ltd. V. Ultravision Techs., LLC*, IPR2021-00190, Paper 7, 14-16 (Jun. 9, 2021).

Further, the attorneys preparing this IPR have not coordinated with Sophos in doing so. *See Twitter, Inc., v. Palo Alto Research Center Inc.*, IPR2021-01458, Paper 11 at 33 (April 6, 2022) (declining to discretionarily deny the petition under *General Plastic* and finding that there was no evidence that Petitioner had coordinated with previous filers).

Accordingly, with no significant relationship between Petitioner and Sophos, this factor heavily weighs against discretionary denial.

2. Factor 2: Whether at the Time of Filing of the First Petition the Petitioner Knew of the Prior Art Asserted in the Second Petition or Should Have Known of It

Because Petitioner is not a petitioner in the Sophos PGR, factor 2 “is of little probative value.” *NetNut*, IPR2021-00465 at 9; *Western Digital Corp. v. Spex Technologies, Inc.*, IPR2018-00084, Paper 14 at 17 (April 25, 2018) (“Because Petitioner has not previously filed a petition against the same patent, factors 2–5 bear little relevance in this case.”).

Regardless, Petitioner became aware of its references during its own prior art search, not from Sophos. The prior art and combinations asserted in the instant Petition under at least Ground 2 are different than those asserted in the Sophos PGR. This factor weighs against discretionary denial.

3. Factor 3: Whether at the Time of Filing of the Second Petition the Petitioner Already Received the Patent Owner's Preliminary Response to the First Petition

Patent Owner has not filed a Patent Owner's Preliminary Response (POPR) in the Sophos PGR. Kaspersky has thus not gained any unfair tactical advantage or road map. See *Code200, UAB, et al. v. Bright Data Ltd.*, IPR2022-00861 and IPR2022-00862, Paper 18 (PTAB Aug. 23, 2022) (Vidal Decision), at 6 (quoting *Cuozzo Speed Techs., LLC v. Lee*, 579 U.S. 261, 272 (2016) (quoting H.R. Rep. No. 112-98, pt. 1, at 45, 48) (hereinafter, "*Code200*").

Factors 1-3 only weigh in favor of discretionary denial when there are "road mapping" concerns under factor 3 or other concerns under factor 2." *Code200*, IPR2022-00861 and IPR2022-00862 at 5. Such road mapping concerns arise where "follow-on petitions would allow petitioners the opportunity to strategically stage their prior art and arguments in multiple petitions, using [the Board's] decisions as a roadmap, until a ground is found that results in the grant of review." *Id.* (citing *General Plastic*, at 17). This factor weighs against discretionary denial.

4. Factors 4 & 5: The Length of Time that Elapsed between the Time the Petitioner Learned of the Prior Art Asserted in the Second Petition and the Filing of the Second Petition, and Whether the Petitioner Provides Adequate Explanation

Because this is Petitioner's *first* petition challenging the '869 patent, the fourth and fifth factors are inapplicable. *See, e.g., Sony Mobile*, IPR2021-00663, 13-15. Moreover, Petitioner became aware of its references during its own prior art search and has been working diligently in preparation of the instant IPR, filed well within the 1-year statutory deadline. 35 U.S.C. § 315(b). Accordingly, these factors weigh "strongly" against discretionary denial. *Sony Mobile* at 14, 15.

5. Factors 6 & 7: The Finite Resources of the Board, and Requirement to Issue a Final Determination within One Year from Institution

While there is a relation between the one-year deadlines for final written decisions on the grounds advanced in Petitioner's IPR petition and Sophos' PGR petition, with Petitioner's IPR petition being filed shortly after Sophos' PGR petition, there is no indication that the Board would be unduly burdened or unable to render a final written decision within the statutory deadline. These factors weigh against discretionary denial or are at least neutral.

C. THE BOARD SHOULD NOT DENY INSTITUTION UNDER 35 U.S.C. § 325(d)

The Board should institute review and should *not* exercise its discretion to deny institution under § 325(d). *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GMBH*, IPR2019-01469, Paper 6, at 6-11 (PTAB Feb. 13, 2020) (precedential).

1. *Advanced Bionics* Step 1 (*Becton* Factors (a), (b), (d)) favors institution

Becton, Dickinson factors (a), (b), and (d) relate to part one of the *Advanced Bionics* framework—whether the art or arguments presented in the Petition are the same or substantially the same as those previously presented to the Office. *Advanced Bionics*, IPR2019-01469 at 10.

Neither primary reference Ideses, nor secondary reference Sharma, was cited during prosecution. Ex. 1001, Cover. In particular, the obviousness over Ideses and combination of Ideses and Sharma are not cumulative to the prior art evaluated during prosecution at least because Ideses discloses the allegedly patentable limitation of “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.” Because the art is not cumulative, the arguments in the present Petition are not substantially the same as the arguments made during prosecution. Petitioner notes that a *different* patent

publication, U.S. Patent Application Pub. No. 2015/0213365 (“Ideses ‘365”), was cited during prosecution of the ‘869 patent, which includes Ianir Ideses as an inventor.

During prosecution, an overlapping portion of Ideses (paragraph ¶0096) and Ideses ‘365 was cited as part of a rejection to dependent claims directed to clustering (“wherein the machine learning model comprises a clustering model”). Ex. 1004, 126. However, Applicant never disputed the teachings of Ideses ‘365 with respect to now-dependent claims 25, 39, and 53 because the Applicant gained allowance solely by amending the independent claims, thereby never developing that portion of the record.

Factors (a), (b), and (d) favor institution.

2. *Advanced Bionics* Step 2 (*Becton* Factors (c), (e), (f)) favors institution

Becton, Dickinson factors (c), (e), and (f) relate to part two of the *Advanced Bionics* framework—whether the Office committed a material error in its prior consideration of the art or arguments. *Advanced Bionics*, IPR2019-01469 at 10.

Although an overlapping portion of Ideses and Ideses ‘365 was cited as part of a rejection, factor (c) favors institution because the Applicant never disputed the teachings of Ideses ‘365 with respect to dependent claims 25, 39, and 53. Specifically, the Examiner rejected then-claims 29 and 44 as obvious over a

combination of an NPL to Shabtai et al., US 2010/0082642 to Forman et al., and Ideses '365. Ex. 1004, 126.

With respect to the dependent claims, Applicant simply stated that the “[t]he Examiner indicated that [certain claims] include allowable subject matter.” Ex. 1004, 127. “Accordingly, to expedite prosecution, the independent claims are amended herein to recite ‘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,’ which is consistent with the proposed amendment discussed during the Examiner interview.” *Id.* “Applicant therefore requests withdrawal of the rejection and allowance of the claims.” *Id.*

Applicant did not address the portions of Ideses '365 identified by the Examiner as disclosing claims 25, 39, and 53, nor did Applicant specifically dispute that such portions failed to disclose the limitations recited in claims 25, 39, and 53. Accordingly, Petitioner is not asking the Office to reconsider any arguments with respect to claims 25, 39, and 53, but merely to make determinations consistent with the Examiner’s previous and uncontested finding that the overlapping portion of Ideses '365 with Ideses discloses the features of claims 25, 39, and 53.

Advanced Bionics explains that “if the record of the Office’s previous consideration of the art is not well developed or silent, then a petitioner may show the Office erred by overlooking something persuasive under factors (e) and (f).” *Advanced Bionics* at 10. Here, the prosecution record is not well developed as to whether the overlapping portion of Ideses ‘365 with Ideses teaches dependent claims 25, 39, and 53. Applicant failed to develop that portion of the record when it chose to advance prosecution exclusively on amendments to the independent claims. Ex.1002, 63. Nothing in the file history suggests that there was a meaningful dispute as to the relevance of the overlapping portion of Ideses ‘365 with Ideses with respect to claims 25, 39, and 53.

With respect to factor (f), Patent Owner obtained allowance of the ‘869 patent after the independent claims were amended to recite, in pertinent part, “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,” features which Ideses discloses. Ex. 1004, 118. The Examiner also did not have the benefit of the detailed expert testimony of V.S. Subrahmanian, Ph.D., and the further evidence of record. Factors (c), (e), and (f) favor institution.

XI. CONCLUSION

Petitioner asks that the Board institute review and cancel each challenged claim.

Dated: June 29, 2023

Respectfully submitted,

By: /Joseph T. Miotke/
Joseph T. Miotke, Reg. No. 47,798
Dewitt LLP
901 Marquette Avenue
2100 AT&T Tower
Minneapolis, MN 55402
Tel: (612) 305-1400

CERTIFICATE OF COMPLIANCE UNDER 37 C.F.R. § 42.24(d)

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 12,438. Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: June 29, 2023

Respectfully submitted,

By: /Joseph T. Miotke/
Joseph T. Miotke, Reg. No. 47,798
Dewitt LLP
901 Marquette Avenue
2100 AT&T Tower
Minneapolis, MN 55402
Tel: (612) 305-1400

CERTIFICATE OF SERVICE

The undersigned certifies that, in accordance with 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105, service was made on Patent Owner as detailed below.

Date of service June 29, 2023

Manner of service FEDERAL EXPRESS

Documents served Petition for Inter Partes Review Under 35 U.S.C. § 312
and 37 C.F.R. § 42.104 of U.S. 11,409,869; Petitioner's
Exhibit List; Exhibits 1001-1037; and Petitioner's
power of attorney.

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

Dated: _____

Respectfully submitted,

By: /Joseph T. Miotke/
Joseph T. Miotke, Reg. No. 47,798
Dewitt LLP
901 Marquette Avenue
2100 AT&T Tower
Minneapolis, MN 55402
Tel: (612) 305-1400