UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner

v.

SKYSONG INNOVATIONS, LLC,
Patent Owner

Inter Partes Review Case No. IPR2025-01398
U.S. Patent No. 11,275,900

DECLARATION OF EREZ ZADOK, Ph.D.

TABLE OF CONTENTS

I. EDUCATIONAL BACKGROUND AND QUALIFICATIONS 11

II. METHODOLOGY AND MATERIALS CONSIDERED 22

III. LEGAL FRAMEWORK 26

IV. OPINION 33

A. THE '900 PATENT 33
B. THE '900 PATENT'S PROSECUTION 34
C. CLAIM CONSTRUCTION 37
D. LEVEL OF A PERSON OF ORDINARY SKILL IN THE ART 38

V. BACKGROUND OF THE TECHNOLOGY 39

A. HIERARCHICAL CLASSIFICATION AND PARENT/CHILD RELATIONSHIPS 39
B. FEATURE EXTRACTION 42
1. Bag-of-Words Representation 42
2. N-gram Representation 43
C. THRESHOLD CLASSIFICATION TECHNIQUE 43
D. CONVOLUTIONAL NEURAL NETWORKS FOR HIERARCHICAL CLASSIFICATION 44
E. DATA PREPROCESSING 45
1. Tokenization 45
2. Stemming and Lemmatization 47
F. DARK WEB CRAWLERS 48

VI. OPINION: GROUND 1 50

A. OBVIOUSNESS OVER DELIU AND PENDAR 50
1. Overview of Deliu 50
2. Overview of Pendar 53
3. Claim 1(Pre): A computer-implemented system for improving classification of criminal activities arising from cyber environments, comprising: 56
4. Claim 1(a): a processor configured to: 56
5. Claim 1(a)(1): access a ground truth dataset generated from deep web forum information, the ground truth dataset defining a predetermined tag hierarchy including predetermined tag labels corresponding topics of the deep web forum information; 57

a. Motivation to Combine Deliu with Pendar (Label Hierarchies) 65
6. Claim 1(a)(2): access data related to a discussion topic for classification from a deep web forum;....69
7. Claim 1(a)(3): extract a set of features from the data by assigning word vectors or paragraph vectors to the data; and 69
8. Claim 1(a)(4): apply a machine classifier to the set of features to generate a prediction list of tags for classifying the discussion topic, wherein the prediction list includes a prediction probability value for each tag of the plurality of tags; and 71
a. Motivation to Combine Deliu with Pendar (Multiple Labels)....75
9. Claim 1(a)(5): add all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value....79
a. Motivation to Combine Deliu with Pendar (Parent/Child Tags) 83
B. CLAIM 2: THE COMPUTER-IMPLEMENTED SYSTEM OF CLAIM 1, WHEREIN A PREDETERMINED TAG HIERARCHY IS DIVIDED INTO PARENT TAGS AND CHILD TAGS, WHEREIN A TOPIC REPRESENTED BY A CHILD TAG IS REPRESENTATIVE OF A SUBSET OF A TOPIC REPRESENTED BY A PARENT TAG, WHEREIN EACH CHILD TAG CORRESPONDS TO A PARTICULAR PARENT TAG, AND EACH TAG IS ASSIGNED A CORRESPONDING PROBABILITY VALUE. 85
C. CLAIM 4: THE COMPUTER-IMPLEMENTED SYSTEM OF CLAIM 1, WHEREIN THE PROCESSOR IS CONFIGURED TO ADD ALL PARENT TAGS OF THE PLURALITY OF TAGS CORRESPONDING TO A CHILD TAG IF A PROBABILITY VALUE ASSOCIATED WITH THE PARENT TAG IS ABOVE THE FIRST PREDETERMINED THRESHOLD VALUE. 88
D. CLAIM 5: THE COMPUTER-IMPLEMENTED SYSTEM OF CLAIM 1, WHEREIN THE SET OF FEATURES ARE EXTRACTED USING TEXTUAL FEATURE EXTRACTION TO IDENTIFY TAGS ASSOCIATED WITH TOPIC TITLES OF THE DATA. 90
E. CLAIM 10: THE COMPUTER-IMPLEMENTED SYSTEM OF CLAIM 1, WHEREIN THE SET OF FEATURES ARE EXTRACTED FROM THE DATA USING A NEURAL NETWORK, WHEREIN THE NEURAL NETWORK IS CONFIGURED TO LEARN A CONTINUOUS DISTRIBUTED VECTOR

REPRESENTATION FOR DOCUMENTS AND WORDS WITHIN THE DATA. 92
F. CLAIM 12 98
1. Claim 12(Pre): A method, comprising: 98
2. Claim 12(a): configuring a processor for executing operations including: 98
3. Claim 12(a)(1): accessing data associated with a deep web forum, the data defining a topic for classification; 98
4. Claim 12(a)(2): extracting a set of features from the data as inputs for a machine classifier; and 99
5. Claim 12(a)(3): apply a machine classifier to the set of features to generate a prediction list of tags for classifying the topic, wherein the prediction list includes a prediction probability value for each tag of the plurality of tags; and 99
6. Claim 12(a)(4): adding all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value. 99
G. CLAIM 13: THE METHOD OF CLAIM 12, FURTHER COMPRISING: OUTPUTTING AN UPDATE TO THE PREDICTION LIST OF TAGS, A TOTAL NUMBER OF THE TAGS OF THE PREDICTION LIST OF TAGS BEING DIFFERENT FROM THE UPDATE TO THE PREDICTION LIST OF TAGS TO MAINTAIN A PREDETERMINED TAG HIERARCHY. 100
VII. OPINION: GROUND 2 102
A. OBVIOUSNESS OVER DELIU, PENDAR, AND FRANK 102
1. Overview of Frank 102
2. Claim 3: The computer-implemented system of claim 1, wherein the processor is configured to remove all child tags of the plurality of tags corresponding to a parent tag from the prediction list of tags where a probability value associated with the child tag is below a second predetermined threshold value. 105
a. Motivation to Combine Deliu-Pendar with Frank 106
VIII. OPINION: GROUND 3 109
A. OBVIOUSNESS OVER DELIU, PENDAR, RAMÍREZ-CORONA 109
1. Overview of Ramírez-Corona 109

2. Claim 3: The computer-implemented system of claim 1, wherein the processor is configured to remove all child tags of the plurality of tags corresponding to a parent tag from the prediction list of tags where a probability value associated with the child tag is below a second predetermined threshold value. 109
a. Motivation to Combine Deliu-Pendar with Ramírez-Corona 111

IX. OPINION: GROUND 4 114

A. OBVIOUSNESS OVER DELIU, PENDAR, AND CHAWLA 114
1. Overview of Chawla 114
2. Claim 6: The computer-implemented system of claim 1, wherein a set of synthetic data samples are created for addition to the ground truth dataset to address class imbalance by creating a set of synthetic sample points along a line defined between a first minority feature vector and a second feature vector and adding the set of synthetic sample points to the ground truth dataset. 115
a. Motivation to Combine Deliu-Pendar with Chawla (Synthetic Data Samples) 117
3. Claim 14: The method of claim 12, further comprising generating a predetermined tag hierarchy based on a ground truth dataset, and leveraging semi-supervised learning to address class imbalance and increase accuracy of the machine classifier. 120
a. Motivation to Combine Deliu-Pendar with Chawla (Tag Hierarchy) 121

X. OPINION: GROUND 5 121

A. OBVIOUSNESS OVER DELIU, PENDAR, AND XIAO 121
1. Overview of Xiao 121
2. Claim 7: The computer-implemented system of claim 1, wherein the ground truth dataset is supplemented by, for every sample of the ground truth dataset in a minority class, adding top similar documents from the deep web forum information, as identified using an elastic search similarity score, and adding them to the ground truth dataset. 122
a. Motivation to Combine Deliu-Pendar with Xiao 124

XI. OPINION: GROUND 6 127

A. OBVIOUSNESS OVER DELIU, PENDAR, AND UKRAINCZYK 127

1. Overview of Ukrainczyk 127

2. Claim 8: The computer-implemented system of claim 1, wherein the data is parsed and segmented into tokens, and predetermined characters and predetermined common words are removed from the data. 127

a. Motivation to Combine Deliu-Pendar with Ukrainczyk 130

3. Claim 9: The computer-implemented system of claim 1, wherein words within the data are reduced to their root form using stemming and lemmatization procedures. 133

a. Motivation to Combine Deliu-Pendar with Ukrainczyk 135

XII. OPINION: GROUND 7 136

A. OBVIOUSNESS OVER DELIU, PENDAR, AND NUNES 136

1. Overview of Nunes 137

2. Claim 11: The computer-implemented system of claim 1, further comprising: a crawler in operable communication with the processor and operable to traverse a plurality of deep web websites to retrieve Internet documents forming at least a portion of the deep web forum information. 137

a. Motivation to Combine Deliu-Pendar with Nunes 139

XIII. CONCLUSION 143

Challenged Claims

Claim Designation	Claim Language
Claim 1(Pre)	A computer-implemented system for improving classification of criminal activities arising from cyber environments, comprising:
Claim 1(a)	a processor configured to:
Claim 1(a)(1)	access a ground truth dataset generated from deep web forum information, the ground truth dataset defining a predetermined tag hierarchy including predetermined tag labels corresponding topics of the deep web forum information;
Claim 1(a)(2)	access data related to a discussion topic for classification from a deep web forum;
Claim 1(a)(3)	extract a set of features from the data by assigning word vectors or paragraph vectors to the data; and
Claim 1(a)(4)	apply a machine classifier to the set of features to generate a prediction list of tags for classifying the discussion topic, wherein the prediction list includes a prediction probability value for each tag of the plurality of tags; and
Claim 1(a)(5)	add all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value.
Claim 2	The computer-implemented system of claim 1, wherein a predetermined tag hierarchy is divided into parent tags and child tags, wherein a topic represented by a child tag is representative of a subset of a topic represented by a parent tag, wherein each child tag corresponds to a particular parent tag, and each tag is assigned a corresponding probability value.

Claim Designation	Claim Language
Claim 3	The computer-implemented system of claim 1, wherein the processor is configured to remove all child tags of the plurality of tags corresponding to a parent tag from the prediction list of tags where a probability value associated with the child tag is below a second predetermined threshold value.
Claim 4	The computer-implemented system of claim 1, wherein the processor is configured to add all parent tags of the plurality of tags corresponding to a child tag if a probability value associated with the parent tag is above the first predetermined threshold value.
Claim 5	The computer-implemented system of claim 1, wherein the set of features are extracted using textual feature extraction to identify tags associated with topic titles of the data.
Claim 6	The computer-implemented system of claim 1, wherein a set of synthetic data samples are created for addition to the ground truth dataset to address class imbalance by creating a set of synthetic sample points along a line defined between a first minority feature vector and a second feature vector and adding the set of synthetic sample points to the ground truth dataset.
Claim 7	The computer-implemented system of claim 1, wherein the ground truth dataset is supplemented by, for every sample of the ground truth dataset in a minority class, adding top similar documents from the deep web forum information, as identified using an elastic search similarity score, and adding them to the ground truth dataset.
Claim 8	The computer-implemented system of claim 1, wherein the data is parsed and segmented into tokens, and predetermined characters and predetermined common words are removed from the data.
Claim 9	The computer-implemented system of claim 1, wherein words within the data are reduced to their root form using stemming and lemmatization procedures.

Claim Designation	Claim Language
Claim 10	The computer-implemented system of claim 1, wherein the set of features are extracted from the data using a neural network, wherein the neural network is configured to learn a continuous distributed vector representation for documents and words within the data.
Claim 11	The computer-implemented system of claim 1, further comprising: a crawler in operable communication with the processor and operable to traverse a plurality of deep web websites to retrieve Internet documents forming at least a portion of the deep web forum information.
Claim 12(Pre)	A method, comprising:
Claim 12(a)	configuring a processor for executing operations including:
Claim 12(a)(1)	accessing data associated with a deep web forum, the data defining a topic for classification;
Claim 12(a)(2)	extracting a set of features from the data as inputs for a machine classifier; and
Claim 12(a)(3)	apply a machine classifier to the set of features to generate a prediction list of tags for classifying the topic, wherein the prediction list includes a prediction probability value for each tag of the plurality of tags; and
Claim 12(a)(4)	adding all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value.
Claim 13	The method of claim 12, further comprising: outputting an update to the prediction list of tags, a total number of the tags of the prediction list of tags being different from the update to the prediction list of tags to maintain a predetermined tag hierarchy.

Claim Designation	Claim Language
Claim 14	The method of claim 12, further comprising generating a predetermined tag hierarchy based on a ground truth dataset, and leveraging semi-supervised learning to address class imbalance and increase accuracy of the machine classifier.

I, Erez Zadok, Ph.D, declare as follows:

1. I am over the age of 21 and am competent to make this declaration. I make this declaration based on facts and matters within my own knowledge and on information provided to me by others, and, if called as a witness, I could and would competently testify to the matters set forth herein.

I. EDUCATIONAL BACKGROUND AND QUALIFICATIONS

2. My *curriculum vitae*, includes a more detailed summary of my background, experience and publications, is available as EX1030.

3. I am a Professor in the Computer Science Department at Stony Brook University (part of the State University of New York ("SUNY") system). I direct the File-systems and Storage Lab (FSL) at Stony Brook's Computer Science Department. My research interests include file systems and storage systems, operating systems, transactional systems including database technologies, information technology and system administration, security/privacy and information assurance, networking, energy efficiency, performance and benchmarking, virtualization, cloud systems, compilers, applied machine learning, and software engineering.

4. I studied at a professional high school in Israel, focusing on electrical engineering ("EE"), and graduated in 1982. I spent one more year at the high school's college division, receiving a special Certified Technician's degree in EE. I

then went on to serve in the Israeli Defense Forces for three years (1983–1986). I received my Bachelor of Science degree in computer science ("CS") in 1991, my Master's degree in CS in 1994, and my PhD in CS in 2001—all from Columbia University in New York.

5. When I began my undergraduate studies at Columbia University, I also started working as a student assistant in the various campus-wide computer labs, eventually becoming an assistant to the head labs manager, who was managing all public computer labs on campus. During that time, I also became more involved with research within the CS Department at Columbia University, conducting research on operating systems, file and storage systems, distributed and networked systems, security, and other topics. I also assisted the CS department's computer administrators in managing the department's computers, which included storage, IT, networking, and cyber-security related duties.

6. In 1991, I joined Columbia University's CS department as a full-time systems administrator, studying towards my MS degree part-time. My MS thesis topic is related to file system reliability, fault tolerance, replication, and failover in mobile networked storage systems using file virtualization. My main duties as a systems administrator involved installing, configuring, and managing many networked servers, proxies, and desktops running several operating systems, as well as network devices setup; this included many software and hardware upgrades,

device upgrades, and BIOS firmware/chipset updates/upgrades. My duties also included ensuring reliable, secure, authenticated access to networked systems/storage and licensed software, as well as software updates, security and bug fixes. Examples of servers and their protocols included email (SMTP), file transfer (FTP), domain names (DNS), network file systems (NFS), network news systems (NNTP), and Web (HTTP).

7. In 1994, I left my systems administrator position to pursue my doctoral studies at Columbia University. My PhD thesis topic was on versatile file system development using stackable (virtualized) file systems, with examples in the fields of security and encryption, efficiency, reliability, and failover. I continued to work part-time as a systems administrator at the CS department, and eventually I was asked to serve as manager to the entire information technology ("IT") staff. From 1991 to 2001, I was also a member of the faculty-level Facilities Committee that oversaw all IT operations at the CS department.

8. As part of my PhD studies at Columbia, I collaborated on projects to develop advanced AI-like techniques to detect previously unknown viruses (a.k.a. "zero-day malware"), using data mining and rule-based detection. This work led to several highly cited papers (over 1,700 citations for one of the papers alone) and two patents. I also became a Teaching Assistant ("TA") for a first-ever Computer

Security course given at Columbia University's CS department with Dr. Matt Blaze as instructor.

9. From 1990 to 1998, I consulted for SOS Corporation and HydraWEB Technologies as a systems administrator and programmer, managing data storage use and backup/restore duties, databases, web servers, as well as information assurance and cyber-security (*e.g.*, malware protection, software licensing). From 1994 to 2000, I led projects at HydraWEB Technologies and then became the Director of Software Development—overseeing the development of several products and appliances such as stateful firewalls and HTTP load-balancers, utilizing network-virtualization and high-availability techniques. From 2009 to 2019, I have consulted for Packet General Networks, a startup specializing in secure, virtualized, network storage and applications' data security in the cloud.

10. In 2001, I joined the faculty of Stony Brook University, a position I have held since that time. In 2002, I joined the Operations Committee, which oversees the IT operations of the CS department at Stony Brook University. From 2006 to 2010, I was the Director of IT Operations of the CS department. My day-to-day duties included setting policies regarding computing, hiring and training new staff, assisting any staff with topics of my specialty, defining requirements for new software/hardware, and purchasing. From 2010 to 2015, I had served as the Co-Chair to the Operations Committee. From 2016 to 2019, I oversaw the IT Operations

as the Chair of the Operations Committee. A significant component of these duties included defining and helping implement policies for data management, to ensure the security of users and their data, and data reliability and availability, while minimizing the inconvenience and performance impact to users. I personally helped setup and maintain an initial virtual-host infrastructure in the department. Since late 2019, I've been a member of the department's Executive Committee that also oversees all IT operations.

11. In 2017, I became the department's Graduate Academic Adviser, advising all Master students (over 400 annually on average) and many other graduate students on an assortment of academic matters. In 2024, I took over as the department's Graduate Program Director, overseeing the entire graduate CS program (700-800 students annually on average).

12. Since 2001, I have personally configured and managed my own research lab's network. This includes setting up and configuring multiple storage systems (*e.g.*, NFS, CIFS/SMB, NAS), virtual and physical environments, applications such as database (*e.g.*, MySQL, PostgreSQL), Web servers (*e.g.*, Apache), and mail servers; user access control (*e.g.*, NIS, LDAP), backups and restores, snapshot policies, and more. I have personally installed, configured, changed, replaced parts, and upgraded components in numerous devices including mobile devices, laptops, desktops, and servers, both physical and virtual.

13. Since 1995, I have taught courses on operating systems, storage and file systems, advanced systems programming in Unix/C, systems administration, data structures, data/software security, and more. My courses often use storage, file systems, distributed systems, and system/network security as key teaching principles and practical examples for assignments and projects. I have taught these concepts and techniques to my students, both to my direct advisees as well as in my courses. For example, in my graduate Operating Systems course, I often cover Linux's kernel mechanisms to protect users, applications, and data files, virtual file systems, as well as distributed storage systems (*e.g.*, NFS). And in the System Administration undergraduate course, I covered many topics such as networking, storage, backups, and configuring complex applications such as mail, web, and database servers.

14. My research often investigates computer systems from many angles: security, efficiency, energy use, scalability, reliability, portability, survivability, usability, ease-of-use, versatility, flexibility, and more. My research gives special attention to balancing five often-conflicting aspects of computer systems: performance, reliability, energy use, security, and ease-of-use.

15. Since joining Stony Brook University in 2001, my group in the File-systems and Storage Lab ("FSL") has developed many file systems and operating system extensions; examples include a highly-secure cryptographic file system, a portable copy-on-write ("COW") versioning file system, a tracing file system useful

to detect intrusions, a replaying file system useful for forensics, a snapshotting and sandboxing file system, a namespace unification file system (that uses stackable, virtualized, file-based COW), an anti-virus file system, an integrity-checking file system, a load balancing and replication/mirroring file system, network file system extensions for security and performance, distributed secure cloud-based storage systems, transactional key-value stores and file systems, OS-level embedded databases, a compiler to convert user-level C code to in-kernel efficient yet safe code, GCC plugins, stackable file system templates, and a Web-based backup system. Many of these projects used one form of virtualization or another (storage, network, host, etc.). I continue to maintain and release newer versions of some of these file systems and software.

16. I have published over 120 refereed publications (in ACM, IEEE, USENIX, and more). To date, my publications have been cited more than 10,000 times (as per Google Scholar as of August 19, 2025). My papers cover a wide range of related technologies such as file systems, storage systems, transactional systems, security, clouds and virtualization, performance benchmarking and optimization, energy efficiency, system administration, web systems, and more. I also published a book titled "Linux NFS and Automounter Administration" (Sybex, 2001), covering systems administration topics related to network storage and data security.

17. Some of my research has led to public software releases that have been used worldwide. I have publicly maintained the Amd Berkeley Automounter in a package called "am-utils" since 1992; this software helps administrators manage the multitude of file system mounts on dozens of different Unix systems, especially helping to automate access to multiple NFS/NAS storage volumes. Since 1997, I have maintained and released several stackable (virtualized) file system software projects for Linux, FreeBSD, and/or Sun Solaris, in a package called FiST. One of my stackable file system encryption projects, called Cryptfs, became the basis for IBM's public release of eCryptfs, now part of Linux. Packet General Networks, for whom I have provided consulting services between 2009 and 2019, licensed another encryption file system called Ncryptfs. Another popular file system released in 2003, called Unionfs, offers virtual namespace unification, transparent shadow copying (a.k.a. copy-on-write or COW), file system snapshotting (*e.g.*, useful for forensics and disaster recovery), and the ability to save disk space by sharing a read-only copy of data across several computers, among other features.

18. My research and teaching make extensive use of data security features. For example, each time I taught the graduate operating system course, the first homework assignment includes the creation of a new system call that performs new or added functionality, often for encrypting a file or verifying its integrity; many of my other assignments cover topics of user/process access control, anti-virus filtering,

and more. Since 2001, over 1,000 graduate students were exposed to these principles directly through my teaching and research at Stony Brook University.

19. Moreover, in an undergraduate course titled "Advanced Systems Programming in Unix/C," I cover many topics of system security and vulnerabilities, such as the structure of UNIX processes, and memory segments such as the heap and stack. This course covers details of several hundred Linux system calls. Often, the first assignment for this course is to develop a tool to encrypt/decrypt files using advanced ciphers, use digital signatures to certify the cipher keys used, and reliably recover files in case of failures. Since 2001, several hundred undergraduate students were exposed to these principles directly through my teaching and research at Stony Brook University.

20. In another undergraduate course, System Administration, I taught network configuration, security, and storage configuration and reliability. In a special topics course on Storage Systems, I covered many topics such as data deduplication, RAID, transactional storage, storage hardware including modern Flash based ones, virtual storage, backup/restore, snapshots and continuous data protection ("CDP"), NAS and SAN, and NFS.

21. Overall, in addition to the aforementioned experience, my technical experience relevant to the '900 Patent at the time of the alleged invention included the following: I studied, researched, developed code, published, taught, released

software, and/or worked on cases covering the following techniques and technologies: written scripts to crawl and harvest web and FTP sites regulaely, including requiring specialized authentiation credentials; develop ML/AI-based meta-heristic techniques to classify data; develop hierarchical databases; studied NLP-based techniques; monitor discussion forums, both public and private, for vulnerability reports, test such reports in confined/safe settings, and assess the risk level of each vulnerability; and developed and patented AI-based techniques to detect zero-day malware, which included collecting and analyzing malware from "dark net" and public sources.

22. My research has been supported by many federal and state grants as well as industry awards, including an NSF CAREER award, two IBM Faculty awards, two NetApp Faculty awards, a Western Digital award, a Facebook award, several Dell-EMC awards, and several equipment gifts. I received the 2008 SUNY Chancellor's Excellence in Teaching award, and the 2022 SUNY Chancellor's Award for Excellence in Scholarship and Creative Activities (both awards can be given only once in a lifetime). In 2021, I was named an ACM Distinguished Member for "Outstanding Scientific Contributions to Computing."

23. My service record to the community includes serving as the co-chair for the USENIX Annual Technical Conference in 2020 (ATC'20); serving as the co-chair for USENIX File and Storage Technologies (FAST'15) in 2015 and on the

FAST Conference Steering Committee from 2015 to 2023; serving on the ACM HotStorage Steering Committee since 2021; and serving as the co-chair in 2012 and on the Steering Committee of the ACM SYSTOR conference since 2012. I have served as an Associate Editor to the ACM Transactions on Storage ("TOS") journal from 2009 to 2022; in 2022, I was named the Editor-in-Chief for ACM's TOS journal.

24. I am a named inventor on four patents, two titled "Systems and Methods for Detection of New Malicious Executables" (U.S. Patent No. 7,487,544, issued February 3, 2009; and U.S. Patent No. 7,979,907, issued July 12, 2011); and two more titled "Multi-Tier Caching," (U.S. Patent No. 9,355,109, issued May 31, 2016; and U.S. Patent 9,959,279, issued May 1, 2018).

25. I have been disclosed as a testifying expert in 24 cases (including *inter partes* review ("IPR") proceedings) in the past four years. I have been deposed 14 times and testified in trial twice.

26. I have personal knowledge of the facts and opinions discussed in this declaration and believe them to be true. If called upon to do so, I would testify competently thereto. I have been warned that willful false statements and the like are punishable by fine or imprisonment, or both. I have had no contact with any of the named inventors of the '900 Patent.

27. My consulting company, Zadoks Consulting Services, is being compensated for my time at my current standard consulting rate. I am also being reimbursed for expenses that I may incur during the course of this work. My compensation is not contingent upon the results of my study and analysis, the substance of my opinions, or the outcome of any proceeding involving the Challenged Claim(s).

28. I have no financial interest in the Petitioner. I have no financial interest in the outcome of this matter or in any litigation involving the '900 Patent.

II. METHODOLOGY AND MATERIALS CONSIDERED

29. I have relied upon my education, knowledge and experience with the technology of communication networks, as well as the other materials as discussed in this declaration in forming my opinions.

30. For this work, I have been asked to review U.S. Patent No. 11,275,900 ("'900 Patent") (EX1001) including the specification and claims, and the '900 Patent's prosecution history ("'900 File History") (EX1002). In developing my opinions relating to the '900 Patent, I have considered the materials cited or discussed herein, including those listed below.

Exhibit	Description
1001	U.S. Patent No. 11,275,900 ("the '900 Patent")
1002	File History of the '900 Patent ("File History")

Exhibit	Description
1004	Isuf Deliu, *Extracting Cyber Threat Intelligence From Hacker Forums,* NTNU Norwegian University of Science and Technology, Doctoral Thesis Submitted June 2017, Carl Leichter, and Katrin Franke ("Deliu")
1005	U.S. Patent No. 9,330,167 to Pendar ("Pendar")
1006	Declaration of June Munford ("Munford Dec.")
1007	Eibe Frank, *Pruning Decision Trees and Lists*, University of Waikato, Submitted January 2000 ("Frank")
1008	Mallinali Ramirez-Corona *et al., Multi-label Classification for Tree and Directed Acyclic Graphs Hierarchies*, Probabilisitic Graphical Models, PGM 2014, (2014) ("Ramírez-Corona")
1009	Nitesh V. Chawla *et al*., *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research 16 (2002) 321-357 ("Chawla")
1010	Tong Xiao *et al*., *KNN and Re-ranking Models for English Patent Mining at NTCIR-7*, Proceedings of NTCIR-7 Workshop Meeting, (2008) ("Xiao")
1011	U.S. Patent No. 7,028,250 to Ukrainczyk *et al*. ("Ukrainczyk")
1012	Eric Nunes *et al*., *Darknet and Deepnet Mining for Proactive Cybersecurity Threat Intelligence*, Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics ("Nunes")
1013	*Top 10 Dark Web and Deep Web Forums of 2025*, https://cyble.com/knowledge-hub/top-10-dark-web-forums/, ("Cyble")
1014	*Dark Web vs. Deep Web: 5 Key Differences*, https://www.Spiceworks.com/it-security/security-general/articles/dark-web-vs-deep-web/, ("Spiceworks")
1015	Thomas Hoffman *et al*., *Hierarchical Document Categorization with Support Vector Machines*, CIKM '04: Proceedings of the thirteenth

Exhibit	Description
	ACM international conference on Information and knowledge management, (2004) ("Hoffman")
1016	Isuf Deliu, Carl Leichter, and Katrin Franke, *Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks*, 2017 IEEE International Conference on Big Data (BIGDATA), 3648-3656 (2017) ("Deliu Paper")
1017	Xinqi Zhu *et al*., *B-CNN: Branch Convolutional Neural Network for Hierarchical Classification, arXiv.org*, (2017) ("Zhu")
1018	Zhicheng Yan *et al*., *HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition*, 2015 IEEE International Conference on Computer Vision (ICCV), (2015) ("Yan")
1019	Quoc Le *et al*., *Distributed Representations of Sentences and Documents*, Proceedings of the 31st International Conference on Machine Learning, PMLR 32(2):1188-1196, (2014) ("Le")
1020	Revanth Patil *et al*., *Multi-class and Multi-label classification of Darkweb Data*, Arizona State University, (Approved March 2018) ("Patil Thesis")
1021	Alex Smola *et al*., *Introduction to Machine Learning*, (2008) ("Smola")
1022	Okapi-BM25, https://en.wikipedia.org/wiki/Okapi_BM25, ("Okapi-BM25 Wikipedia Webpage")
1023	M. Hassler *et al*., *Text preparation through extended tokenization*, Data Mining VII: Data, Text and Web Mining and their Business Applications, (2006) ("Hassler")
1024	Jasmeet Singh *et al*., *A systematic review of text stemming techniques*, Artificial Intelligence Review, (2016) ("Singh")
1025	Tianjun Fu *et al*., *A Focused Crawler for Dark Web Forums*, Journal of the American Society for Information Science and Technology, (2010) ("Fu")

Exhibit	Description
1026	Mitch Macdonald *et al*., *Identifying Digital Threats in a Hacker Web Forum*, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, (2015) ("Macdonald")
1027	Roman Eisner *et al*., *Improving Protein Function Prediction using the Hierarchical Structure of the Gene Ontology*, 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, (2005) ("Eisner")
1028	Alaa Alahmadi, *Automatic text classification using bag of words and bag of concepts based representation*s, University of Limerick, (2016) ("Alahmadi")
1029	Microsoft Computer Dictionary, 5th ed., 2002, excerpts ("Microsoft Computer Dictionary")
1030	Curriculum vitae (CV) of Dr. Erez Zadok.

31. I have considered these materials from the viewpoint of a POSITA as of the priority date of the '900 Patent. For the purposes of this declaration, I have been asked to assume that the priority date of the '900 Patent is May 9, 2018. I note that my opinions provided in this declaration are made from the perspective of a POSITA as of this priority date of the '900 Patent, unless expressly stated otherwise. To the extent that I use any verb tense in this declaration that is present tense (*e.g.*, "a POSITA would understand" instead of "a POSITA would have understood"), such verb tense should be understood to be my opinion as of the '900 Patent's priority date (again, unless expressly stated otherwise). I merely use the present verb tense for ease of reading.

III. LEGAL FRAMEWORK

32. I am a technical expert and do not offer any legal opinions. However, counsel has informed me as to certain legal principles regarding patentability and related matters under United States patent law, which I have applied in performing my analysis and arriving at my technical opinions in this matter.

33. I have been informed that the Patent Trial and Appeal Board ("PTAB") now applies the *Phillips* standard regardless of whether a patent has expired. I have been informed that under the *Phillips* standard, claim terms are to be given the meaning they would have to a person having ordinary skill in the art at the time of the invention, taking into consideration the patent, its file history, and, secondarily, any applicable extrinsic evidence (*e.g.*, dictionary definitions). I understand that, for purposes of this proceeding, Petitioner applies the ordinary and customary meaning of the claim terms as understood by a POSITA. I have applied the plain meaning of the claim language as instructed by *Phillips*.

34. I have also been informed that a person cannot obtain a patent on an invention if the differences between the invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art. I have been informed that a conclusion of obviousness may be founded upon more than a single item of prior art. I have been further informed that obviousness is determined by evaluating the

following factors: (1) the scope and content of the prior art, (2) the differences between the prior art and the claim at issue, (3) the level of ordinary skill in the pertinent art, and (4) secondary considerations of non-obviousness. In addition, the obviousness inquiry should not be done in hindsight. Instead, the obviousness inquiry should be done through the eyes of a person of ordinary skill in the relevant art at the time the patent was filed.

35. In considering whether certain prior art renders a particular patent claim obvious, counsel has informed me that I can consider the scope and content of the prior art, including the fact that one of skill in the art would regularly look to the disclosures in patents, trade publications, journal articles, industry standards, product literature and documentation, texts describing competitive technologies, requests for comment published by standard setting organizations, and materials from industry conferences, as examples. I have been informed that for a prior art reference to be proper for use in an obviousness analysis, the reference must be "analogous art" to the claimed invention. I have been informed that a reference is analogous art to the claimed invention if: (1) the reference is from the same field of endeavor as the claimed invention (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor (even if it is not in the same field of endeavor as the claimed invention). In order for a reference to be "reasonably pertinent" to the problem, it must logically have commended itself

to an inventor's attention in considering his problem. In determining whether a reference is reasonably pertinent, one should consider the problem faced by the inventor, as reflected either explicitly or implicitly, in the specification. I believe that all of the references that my opinions in this IPR are based upon are well within the range of references a person of ordinary skill in the art would consult to address the type of problems described in the Challenged Claims.

36. I have been informed that, in order to establish that a claimed invention was obvious based on a combination of prior art elements, a clear articulation of the reason(s) why a claimed invention would have been obvious must be provided. Specifically, I am informed that a combination of multiple items of prior art renders a patent claim obvious when there was an apparent reason for one of ordinary skill in the art, at the time of the invention, to combine the prior art, which can include, but is not limited to, any of the following rationales: (A) combining prior art methods according to known methods to yield predictable results; (B) substituting one known element for another to obtain predictable results; (C) using a known technique to improve a similar device in the same way; (D) applying a known technique to a known device ready for improvement to yield predictable results; (E) trying a finite number of identified, predictable potential solutions, with a reasonable expectation of success; (F) identifying that known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design

incentives or other market forces if the variations are predictable to one of ordinary skill in the art; or (G) identifying an explicit teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine the prior art references to arrive at the claimed invention.

37. I am informed that the existence of an explicit teaching, suggestion, or motivation to combine known elements of the prior art is a sufficient, but not a necessary, condition to a finding of obviousness. This so-called "teaching suggestion-motivation" test is not the exclusive test and is not to be applied rigidly in an obviousness analysis. In determining whether the subject matter of a patent claim is obvious, neither the particular motivation nor the avowed purpose of the patentee controls. Instead, the important consideration is the objective reach of the claim. In other words, if the claim extends to what is obvious, then the claim is invalid. I am further informed that the obviousness analysis often necessitates consideration of the interrelated teachings of multiple patents, the effects of demands known to the technological community or present in the marketplace, and the background knowledge possessed by a person having ordinary skill in the art. All of these issues may be considered to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent.

38. I also am informed that in conducting an obviousness analysis, a precise teaching directed to the specific subject matter of the Challenged Claim need not be

sought out because it is appropriate to take account of the inferences and creative steps that a person of ordinary skill in the art would employ. The prior art considered can be directed to any need or problem known in the field of endeavor at the time of invention and can provide a reason for combining the elements of the prior art in the manner claimed. In other words, the prior art need not be directed towards solving the same specific problem as the problem addressed by the patent. Further, the individual prior art references themselves need not all be directed towards solving the same problem. I am informed that common sense is important and should be considered. Common sense teaches that familiar items may have obvious uses beyond their primary purposes.

39. I also am informed that the fact that a particular combination of prior art elements was "obvious to try" may indicate that the combination was obvious even if no one attempted the combination. If the combination was obvious to try (regardless of whether it was actually tried) or leads to anticipated success, then it is likely the result of ordinary skill and common sense rather than innovation. I am further informed that in many fields it may be that there is little discussion of obvious techniques or combinations, and it often may be the case that market demand, rather than scientific literature or knowledge, will drive the design of an invention. I am informed that an invention that is a combination of prior art must do more than yield predictable results to be non-obvious.

40. I am informed that for a patent claim to be obvious, the claim must be obvious to a person of ordinary skill in the art at the time of the invention. I am informed that the factors to consider in determining the level of ordinary skill in the art include (1) the educational level and experience of people working in the field at the time the invention was made, (2) the types of problems faced in the art and the solutions found to those problems, and (3) the sophistication of the technology in the field.

41. I am informed that it is improper to combine references where the references teach away from their combination. I am informed that a reference may be said to teach away when a person of ordinary skill in the relevant art, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the patent applicant. In general, a reference will teach away if it suggests that the line of development flowing from the reference's disclosure is unlikely to be productive of the result sought by the patentee. I am informed that a reference teaches away, for example, if (1) the combination would produce a seemingly inoperative device, or (2) the references leave the impression that the product would not have the property sought by the patentee. I also am informed, however, that a reference does not teach away if it merely expresses a general preference for an alternative invention but does

not criticize, discredit, or otherwise discourage investigation into the invention claimed.

42. I am informed that even if a prima facie case of obviousness is established, the final determination of obviousness must also consider "secondary considerations" if presented. In most instances, the patentee raises these secondary considerations of non-obviousness. In that context, the patentee argues an invention would not have been obvious in view of these considerations, which include: (a) commercial success of a product due to the merits of the claimed invention; (b) a long-felt, but unsatisfied need for the invention; (c) failure of others to find the solution provided by the claimed invention; (d) deliberate copying of the invention by others; (e) unexpected results achieved by the invention; (f) praise of the invention by others skilled in the art; (g) lack of independent simultaneous invention within a comparatively short space of time; (h) teaching away from the invention in the prior art.

43. I am further informed that secondary considerations evidence is only relevant if the offering party establishes a connection, or nexus, between the evidence and the claimed invention. The nexus cannot be based on prior art features. The establishment of a nexus is a question of fact. While I understand that the Patent Owner here has not offered any secondary considerations at this time, I will

supplement my opinions in the event that the Patent Owner raises secondary considerations during the course of this proceeding.

IV. OPINION

A. The ’900 Patent

44. The ’900 Patent describes “a computer-implemented system for improving classification of data associated with the deep web or dark net[.]” ’900 Patent (EX1001), Abstract, Fig. 6. Dark web environments provide an enormous repository of criminal data that requires automatic classification. *Id*., 1:17-62. The ’900 Patent, therefore, involves “automatically assigning one or more labels (tags), in a hierarchical structure, to discussion topics seen in d2web [dark web-deep web] forums.” *Id*., 3:11-15. The ’900 Patent refers to classifying d2web forums as a “multiclass, multi-label classification problem.” *Id*., 3:40-43.

45. The data to be classified is naturally hierarchical. *Id*., 5:16-17, 5:63-65. Therefore, the ’900 Patent leverages tag hierarchy to ensure consistency in its data. *Id*., 5:62-6:67. For example, “for a predicted child tag every parent tag should also be predicted.” *Id*., 6:5-7. The ’900 Patent provides three approaches for ensuring hierarchical consistency: adding parent tags, removing child tags, and a combination adding parent and removing child tags. *Id*., 6:24-27. In the example of adding parent tags, “[i]f a child tag is predicted by the classifier and if the corresponding parent tags in the hierarchy are not predicted, then all the parent tags

are added to the prediction list." *Id*., 6:2-35. For removing all child tags, "all predicted child tags are removed that do not have all the hierarchical parent tags." *Id*., 6:34-43. A combination of adding parent and removing child tags is shown in example algorithm 1 below:

Algorithm 1 Combination of Adding Parent Tags and Removing Child Tags (T)

Input: Prediction list of tags (T) for each document.
Output: Prediction list of tags that respect the tag hierarchy for each document.
1: $\beta \leftarrow$ remove child threshold value
2: $\alpha \leftarrow$ add parent threshold value
3: for t:= 1 to T do
4: p(t) $\leftarrow$ prediction probability of tag t.
5: if p(t) $< \beta$ then remove all child tags of tag t from prediction list
6: for t:= 1 to T do
7: p(t) $\leftarrow$ prediction probability of tag t.
8: if p(t) $> \alpha$ then add all parent tags of tag t from prediction list
return updated prediction list

Id., Algorithm 1 (6:45-57).

B. The '900 Patent's Prosecution

46. The '900 Patent claims priority to provisional application 62/668,878, filed May 9, 2018. For the purposes of this declaration, I have been asked to assume that the priority date of the '900 Patent is May 9, 2018.

47. Original claim 1 is shown below for reference:

1. A computer-implemented system for improving classification of criminal activities arising from cyber environments, comprising:

a processor configured to:

access a ground truth dataset generated from deep web forum information, the ground truth dataset defining a predetermined tag hierarchy including predetermined tag labels corresponding topics of the deep web forum information;

access data related to a discussion topic for classification from a deep web forum;

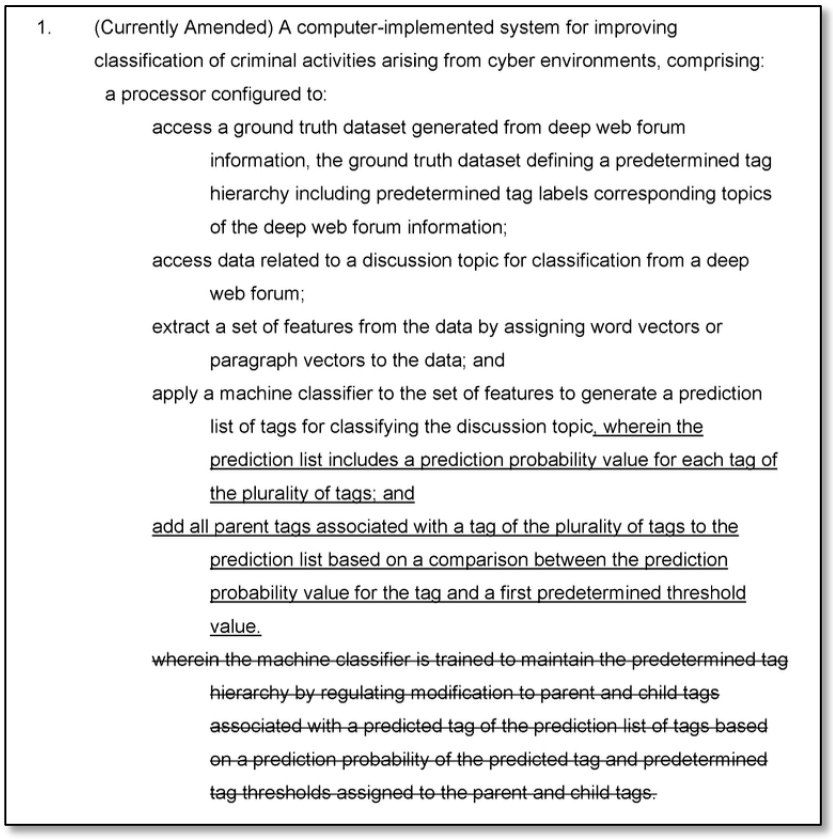
extract a set of features from the data by assigning word vectors or paragraph vectors to the data; and

apply a machine classifier to the set of features to generate a prediction list of tags for classifying the discussion topic,

wherein the machine classifier is trained to maintain the predetermined tag hierarchy by regulating modification to parent and child tags associated with a predicted tag of the prediction list of tags based on a prediction probability of the predicted tag and predetermined tag thresholds assigned to the parent and child tags.

'900 Patent File History (EX1002), 29.

48. In a response to a first office action, Applicant narrowed the claims to require, in relevant part, "the prediction list includes a prediction probability value for each tag of the plurality of tags" and "add all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparisong between the prediction probability value for the tag and a first predetermined threshold value." '900 File History, 284. Amended claim 1 is shown below for reference:



1. (Currently Amended) A computer-implemented system for improving classification of criminal activities arising from cyber environments, comprising:

a processor configured to:

access a ground truth dataset generated from deep web forum information, the ground truth dataset defining a predetermined tag hierarchy including predetermined tag labels corresponding topics of the deep web forum information;

access data related to a discussion topic for classification from a deep web forum;

extract a set of features from the data by assigning word vectors or paragraph vectors to the data; and

apply a machine classifier to the set of features to generate a prediction list of tags for classifying the discussion topic, wherein the prediction list includes a prediction probability value for each tag of the plurality of tags; and

add all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value.

~~wherein the machine classifier is trained to maintain the predetermined tag hierarchy by regulating modification to parent and child tags associated with a predicted tag of the prediction list of tags based on a prediction probability of the predicted tag and predetermined tag thresholds assigned to the parent and child tags.~~

'900 Patent File History (EX1002), 284.

49. Applicant argued "[t]he step of adding **all** parent tags … aims to solve the problem … observed that for some tags returned by the classifier function, the corresponding parent tags [are] not returned" such as when "the tag is wrongly predicted (false positive) or when some of the correct tags (false negatives) are not predicted." *Id.,* 291 (emphasis original). Applicant differentiated prior art that only "evaluates inclusion of a parent tag or a child tag on a **case-by-case basis**[.]"[1] *Id.,* 291-292 ("the present system adds ALL parent tags associated with a child tag if the

[1] All bold emphases in this Declaration were added unless otherwise noted.

child tag has a high enough probability prediction") (bold and all-uppercase emphases original). A notice of allowance subsequently issued.

50. As set forth below, these concepts were well-known in the art before the earliest priority date of the '900 Patent.

C. Claim Construction

51. I have been informed by counsel that the first step in an unpatentability analysis involves construing the claims, as necessary, to determine their scope. Second, the construed claim language is then compared to the disclosures of the prior art. I am informed that claims are generally given their ordinary and custom meaning as understood by one of ordinary skill in the art at the time of the invention in light of the patent specification and prosecution history, *i.e.*, the intrinsic record.

52. I understand that Petitioner proposes the Board correct a typographical error in Claims 3 and 4:

> **Claim 3**
>
> "wherein the processor is configured to remove all child tags of the plurality of tags corresponding to a parent tag from the prediction list of tags where a probability value associated with the ~~child~~ [parent] tag is below a second predetermined threshold value"
>
> **Claim 4**
>
> "wherein the processor is configured to add all parent tags of the plurality of tags corresponding to a child tag if a probability value associated with the ~~parent~~ [child] tag is above the first predetermined threshold value."

53. Accordingly, I have also adopted Petitioner's interpretation of Claims 3 and 4 in my opinions below, and I take no position as to the correctness of these interpretations. For all other claim language, I have applied the plain and ordinary meaning as defined in the preceding paragraph.

D. Level of a Person of Ordinary Skill in the Art

54. I understand that the '900 Patent, the record of proceedings at the Patent Office (which I understand is called the "File History" or "Prosecution History"), and the teachings of the prior art are evaluated from the perspective of a person of ordinary skill in the art ("POSITA") at the time the claimed invention was made. I understand that the factors considered in determining the ordinary level of skill in the art include: (i) the levels of education and experience of persons working in the field; (ii) the types of problems encountered in the field; and (iii) the sophistication of the technology. I may also consider, if available, the education level of the inventor, prior art solutions to the problems encountered in the art, and the rapidity with which innovations are made in the relevant art.

55. I understand that a person of ordinary skill in the art is not a specific real individual, but rather a hypothetical individual having the qualities reflected by the factors above. This hypothetical person has knowledge of all prior art in the relevant field as if it were arranged on a workshop wall and takes from each reference what it would teach to a person having the skills of a POSITA.

56. Based on my review and analysis of the ’900 Patent, the prior art cited herein, and the ordinary skill factors described in this section, a POSITA in the field of the ’900 Patent at the time of the earliest possible priority date (May 9, 2018) would have had a bachelor’s degree in computer science, electrical or computer engineering, or an equivalent degree; and 2-3 years of work or research experience in computer security and networking. Additional education (*e.g.*, MS degree) might substitute for less experience and vice versa.

57. As of May 9, 2018, I had more than ordinary skill in the art and meet at least these minimum qualifications to be a person having ordinary skill in the art as of the time of the claimed invention of the ’900 Patent. I am, however, familiar with the skills and knowledge possessed by those I would have considered to be of ordinary skill in the art as of that date. *See* §I above, Educational Background and Qualifications.

58. My opinions provided in this declaration would not change in view of minor modifications to this level of ordinary skill.

V. BACKGROUND OF THE TECHNOLOGY

A. Hierarchical Classification and Parent/Child Relationships

59. Classification of data based on a hierarchy of classes was well known in the art at the time of the ’900 Patent. For example, Hoffman states, “[a]utomatically categorizing documents into pre-defined topic hierarchies or

taxonomies is a crucial step in knowledge and content management." Hoffman (EX1015), 1. Classifying documents hierarchically is especially important when it comes to large datasets in order to organize and maintain the information properly. *See e.g., id.*, 1 ("Document categorization is a crucial and well-proven instrument for organizing large volumes of textual information"); *see also id.,* 1–2 ("There is reason to believe that taxonomies offer valuable information which learning methods should be able to capitalize on … The potential loss of valuable information suffered from ignoring class hierarchies has been pointed out many times before and has led to a number of approaches that deal with ways to exploit hierarchies").

60. It was known that machine classifiers, such as Support Vector Machines (SVMs), would exploit hierarchical information in data classification. *Id.*, 1 ("In this paper, we propose a novel hierarchical classification method that generalizes Support Vector Machine learning and that is based on discriminant functions that are structured in a way that mirrors the class hierarchy"); *see also id.,* 2 ("…we present a generalization of the SVM classification architecture [] that directly incorporates prior knowledge about class relationships").

61. In hierarchical data classification, the "true path rule" plays an important role in maintaining the consistency of the hierarchy. *See e.g.*, Eisner (EX1027), 2 ("Without the true path rule, predictions within the hierarchy would be inconsistent"). The true path rule is described by Eisner as "propagating the labels

via the is-a relationship to the root node[.]" *Id.,* 4. Meaning, if data is assigned to a certain label, "then it is also assigned the label of all of its ancestors up to the root node." *Id.* Thus, adding the ancestor or parent labels of an assigned label to a data instance was a well-known technique in hierarchical data classification at the time of the '900 Patent.

62. Overall, relationships between objects are well known to software programmers. This is something often taught in the very first computer science class in a bachelor's CS program, as I personally did (*see* §I above). A "parent/child" relationship is one "[p]ertaining to or constituting a relationship between nodes in a tree data structure in which the parent is one step closer to the root (that is, one level higher) than the child." Microsoft Computer Dictionary (EX1029), 4 (definition 2).

63. One of the most popular data structures taught is a "tree," which is "[a] data structure containing zero or more nodes that are linked together in a hierarchical fashion. If there are any nodes, one node is the root; each node except the root is the child of one and only one other node; and each node has zero or more nodes as children. *See also* child (definition 2), graph, leaf, node (definition 3), parent/child (definition 2), root." Microsoft Computer Dictionary (EX1029), 7.

64. Parent/child relationships lend themselves naturally to the creation of complex hierarchies. For example, the database "hierarchical model" is "[a] model used in database management in which each record may be the 'parent' of one or

more child records, which may or may not have the same structure as the parent; a record can have no more than one parent. Conceptually, therefore, a hierarchical model can be, and usually is, regarded as a tree. The individual records are not necessarily contained in the same file. *See also* tree." Microsoft Computer Dictionary (EX1029), 3.

B. Feature Extraction

65. Bag-of-words and n-gram feature extraction were both common vectorization feature-extraction techniques that were well known in the art at the time of the '900 Patent. *See e.g.*, Le (EX1019), 1 ("Perhaps the most common fixed-length vector representation for texts is the bag-of-words or bag-of-n-grams [] due to its simplicity, efficiency and often surprising accuracy").

1. Bag-of-Words Representation

66. Bag-of-words (BOW) is the "most popular representation model used in Information Retrieval (IR)[.]" Alahmadi (EX1028), 51. Bag-of-words "breaks terms into their constituent words, e.g., it breaks 'text classification' into the words 'text' and 'classification[.]'" *Id*., 52. In BOW representation, each dimension in a vector space model represents a single word. *Id*., 51 ("Each dimension in the VSM corresponds to an individual word, and if a given word occurs in the text, based on the type of BOW used, the value of the word could be defined as a binary or frequency vector"). BOW considers the frequency of a word in a "frequency

vector," so "if the word appears in the text the corresponding value is the number of times the word appears in the text[.]" *Id*., 51.

2. N-gram Representation

67. N-gram representation is another commonly used technique in information retrieval. Alahmadi (EX1028), 52-53 ("N-grams were first mentioned in the context of the theory of communication by Shannon in 1948 [65] Since then; N-grams have been widely used in IR applications [66] and ATC [2, 64]."). N-gram representation can consider word-order in a document as "N-gram is a contiguous sequence of N items from a given sequence of text or speech." *Id*. The number of N-grams correlates to the size of the dimensional feature in the vector space model of the document. *See e.g., id.,* 54 (describing "if the length of the N-grams is five or more, the total number of unique N-grams is larger than the total number of words found in the text which leads to high-dimensional feature in VSM []"). Additionally, N-gram representation can deal with noise in textual data. *See e.g., id.,* 53 ("N-gram based representation has the capability of dealing with noise in text such as mistakes caused by human error (spelling for example) or errors generated during the Optical Character Recognition (OCR) process []").

C. Threshold Classification Technique

68. Assigning a label when its score exceeds a threshold was a common and widely used classification technique at the time of the '900 Patent. As an

example of the wide use of threshold values in classification, Ukrainczyk discusses that a document is classified into a category "provided the category score exceeds a predetermined threshold." Ukrainczyk (EX1011), 3:48-50; *see also id.,* 20:45-48 ("The system further includes the capability to specify a threshold score above which a tag score must be for a tag to a node in a particular perspective to be put on a document").

D. Convolutional Neural Networks for Hierarchical Classification

69. It was known at the time of the '900 Patent that Convolutional Neural Networks (CNN) were classifiers that were utilized for hierarchical data classification. For example, Zhu introduces a hierarchical CNN model. *See e.g.,* Zhu (EX1017), 1 ("In this paper, we introduce a special CNN based model **integrated with the prior knowledge of hierarchical category relations** []. We name it *Branch Convolutional Neural Network* (B-CNN) as it contains several branch networks along the main convolution workflow **to do predictions hierarchically**") (italicized emphasis original). *Id.*, 1. Zhu acknowledges that "CNN based models are naturally hierarchical." *Id.* Because of this, a "way to embed a hierarchy of classes into a CNN model is to output multiple predictions along the CNN layers as the data flow through[.]" *Id.* Compared to traditional CNN models, hierarchical CNN models "can do predictions in a more interpretable way

and even boost the final classification as the hierarchical prior is a good guide to the classifier." *Id.*

70. Yan also discusses a CNN model that "leverage[s] the hierarchical structure of categories" in image classification. Yan (EX1018), 1. Specifically, Yan proposes "a generic and principled hierarchical architecture, *Hierarchical Deep Convolutional Neural Network* (HD-CNN), that decomposes an image classification task into two steps." (italicized emphasis original). *Id.* The first step of classification is to separate easy classes using a "coarse category CNN classifier." *Id.* Second, the more challenging classes are routed to a "fine category" CNN classifier. *Id.* This HD-CNN classifier "follows the coarse-to-fine classification paradigm" for hierarchical classification and achieves "lower error than the corresponding building block CNN[.]" *Id.,* 1-2.

E. Data Preprocessing

71. Tokenization, stemming, and lemmatization are known techniques that are used to pre-process textual data to an acceptable format for classification. The use of these pre-processing techniques was known to increase the performance of classifiers.

1. Tokenization

72. Compilers and parsers have long since processed text (*e.g.*, human-written computer code) and broke it up into its constituent components, called

"tokens." I have personally studied and written parsers that tokenize input texts, including as part of my PhD dissertation. *See* §I above. A "token" is "Any nonreducible textual element in data that is being parsed—for example, the use in a program of a variable name, a reserved word, or an operator. Storing tokens as short codes shortens program files and speeds execution. *See also* Basic, parse." Microsoft Computer Dictionary (EX1029), 6 (definition 2). To "parse" is "[t]o break input into smaller chunks so that a program can act upon the information." *Id.*, 5. A "parser" is "[a]n application or device that breaks data into smaller chunks so that an application can act on the information. *See also* parse." *Id*.

73. Hassler describes "tokenization" as "the first step of any kind of natural language text preparation." Hassler (EX1023), 1. The point of tokenization is to "convert a stream of characters into a stream of processing units called tokens." *Id.* The process of tokenization begins with raw text being "segmented into textual units" through cleansing and filtering. *See e.g., id.*, 2 ("First, raw texts are preprocessed and segmented into textual units. This step comprises cleansing and filtering (*e.g.*, whitespace collapsing, stripping extraneous control characters) [] and removal of all kinds of structural or layout relevant markup (*e.g.*, XML tags)"). Hassler describes that following this step, "extended tokenization segments the plain text into appropriate processing units." *Id.,* 2. Hassler highlights that methods performed in pre-processing provide the benefit of simplicity later in the data

classification process. *See e.g., id.,* ("Thus, the **complexity of subsequent processing tasks (*e.g.*, tagging, chunking) is reduced dramatically by early made decisions**").

2. Stemming and Lemmatization

74. Stemming and lemmatization are also language processing techniques that are known to ultimately improve data classification. Stemming is defined as "a procedure in which the various morphological variants of words are matched to their stem (root, base word)." Singh (EX1024), 2. Singh presents the example that the "root word" of the words "maintaining," "maintained," and "maintenance" is "maintain[.]" *Id.*

75. Lemmatization is defined to "perform[] complete morphological analysis of the words to determine the lemma[.]" *Id.,* 3. The "lemma" of a word is the dictionary form of a word. *See e.g., id.,* ("Lemmas, on the other hand, are valid linguistic components and a dictionary form of a lexeme. Lexeme corresponds to a collection of all the word variant forms that have similar meaning"). For example, Singh illustrates that the "lemma" of the words "run," "ran," "runs," and "running" is "run." *Id.*

76. In practice, stemming and lemmatization are considered "sibling processes" as they both "perform a similar function of reducing the variant words in the input text." *Id.* The reduction of the variant of words in textual data can decrease

the size of the input file into a classifier and ultimately increase classifier performance. *See e.g.*, *id.,* 2 ("Secondly, stemming is viewed as a mechanism to reduce the size of the index file since the various variant terms are reduced to a single term[]. Sometimes stemming is found to be so useful, that the size of the index file is reduced to half of the original size").

F. Dark Web Crawlers

77. Web crawlers are techniques that are implemented by software programs to "traverse the World Wide Web information space by following hypertext links and retrieving web documents[.]" Fu (EX1025), 1. Web crawlers that were specifically focused on gathering documents related to a certain topic were known at the time of the '900 Patent. *See e.g., id.,* ("Crawlers can be used for general-purpose search engines **or for topic-specific collection building**"). Further, web crawlers that were focused on collecting documents from dark web forums were known. For example, Fu proposes "the development of a focused crawler that can collect Dark Web forums." *Id.*, 2. Fu describes that there was a need for a web crawler that is focused on and can collect Dark Web forums as the main concern with Dark Web forums is accessibility. *Id.*, 1. Dark Web forums are a part of the "Hidden Web[,]" "which is not easily accessible through normal Web navigation or standard crawling." *Id.*, 2.

78. As an additional example, Macdonald describes using a hacker-forum focused web-crawler. *See e.g.,* Macdonald (EX1026), 1 ("First, **discussion posts were collected from a selected web forum using a customized web-crawler**"). The software of the crawler discussed is called "Open Discussion Forum Crawler" which was customized to capture the content posted in hacker discussion forums. *Id.,* 2. This crawler collected the data from forums by "downloading its webpages," "parsing the page apart" to collect the useful information, and then storing the information in a database that corresponds to the discussion forum. *Id.*

79. It should be noted that there is nothing special about the "dark web" in the context of this patent. Harvesting dark web data may benefit from using custom crawlers, not claimed in this patent (other than claim 11). The same techniques used to analyze publicly available information (*e.g.*, text, html, source code, blog posts, etc.) can be applied with little (if any) changes to data gathered from dark web sources. Modern dark webs may use high-speed networks, the web, and Internet. However, in the late 1980s and 1990s such "dark" information was usually posted on obscure USENET newsgroups as well as disseminated via Bulletin Board Systems (BBSs), and one had to connect via slower dial-up modems.

VI. OPINION: GROUND 1

A. Obviousness over Deliu and Pendar

80. Claims 1, 2, 4, 5, 10, 12, and 13 would have been obvious over Deliu and Pendar.

1. Overview of Deliu

81. I have reviewed the Declaration of Ms. June Munford regarding her methodology in identifying and locating Deliu. Munford Dec. (EX1006), ¶¶6-39. As a person with more than ordinary skill in the art, I agree with Ms. Munford's methodology in identifying and locating Deliu.

82. In my over 30 years in academia, I have found out that theses are serious, formal, peer-reviewed documents that are often deposited in the institutions' libraries and/or graduate schools. This deposit is crucial for preservation and long-term archival—as well as for verification/confirmation of who defended a thesis successfully, when, and the title/content. I have personally searched, downloaded, and reviewed many theses (including when I was in graduate school myself). *See* §I above.

83. Specifically, I have been asked to opine on whether Deliu would have been accessible to person's interested and ordinarily skilled in the subject matter. I have also been asked to consider whether, even without knowledge of the work, the Deliu thesis title is adequately descriptive that an individual interested in this research area would have located it through ordinary means. I understand Deliu is

a Master's thesis presented in 2017 at the Norwegian University of Science and Technology (NTNU). I have been made aware of a Declaration from June Munford that describes how interested individuals could have found Deliu before the priority date of the '900 Patent. Munford Dec. (EX1006), ¶¶6-39. Ms. Munford describes locating Deliu through NTNU's open-access library records. Namely, Ms. Munford details a number of search queries that produced the Deliu thesis in the search results. Munford Dec. (EX1006), ¶32. The search queries include a variety of keywords from the title of the thesis, such as "Extracting Cyber Threat", "Threat Intelligence", "Cyber Threat Intelligence", and "Hacker Forums", as well as the title in full. *Id.* The search queries also produced Deliu using keywords from the Abstract of the thesis, such as "Recent Developments in Deep Learning", "Support Vector Machines", and "Hacker Community". *Id.*

84. I agree with Ms. Munford's methodology, and it is therefore my opinion that a person interested in this subject matter could indeed have searched for and discovered Deliu as of 2017. NTNU is a known research institution in engineering and technology, including in at least the fields of cyber intelligence and machine learning. An individual interested in these fields, even without prior knowledge of Deliu, would have been able to locate the thesis through a reasonable search given the descriptive keywords in Deliu's title or other keywords in the body of the thesis.

85. Deliu compares the performance of machine learning approaches in "Extracting Cyber Threat Intelligence From Hacker Forums[.]" Deliu (EX1004), 1. Deliu obtained a leaked dataset from a hacker forum called *Nulled.IO*. *Id.*, 62. A labeled dataset is constructed by classifying the forum posts into *irrelevant* and *relevant* posts based on whether the posts contain one of a number of hand-selected keywords related to security. *Id.*, 62-63. The *relevant* posts were then further classified into a number of information-security categories. *Id.*, 63,74-76, Fig. 12.

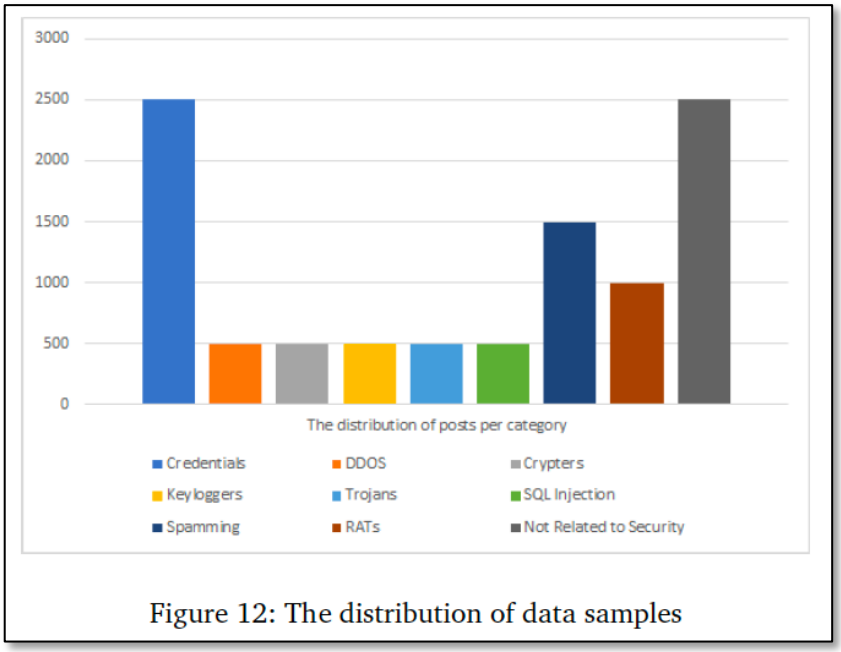


Figure 12: The distribution of data samples

Id., Figure 12.

86. Deliu teaches using well-known machine-learning techniques to classify forum posts. *Id.*, 7 ("In this thesis, we explore the capbailities of Machine Learning methods in the task of locating relevant threat intelligence from hacker forums"). For example, Deliu teaches vectorizing text using Bag-of-Words and n-

gram feature-extraction techniques, and feeding those feature vectors to k-NN, Decision Tree, and SVM classification algorithms. *Id.*, 64-66 ("Traditional Classifiers with Bag-of-Words features … Traditional Classifiers with n-gram features"). Deliu also teaches vectorizing text using word2vec, GloVe, and Random vector feature extraction algorithms for analysis by a Convolutional Neural Network. *Id.*, 66-70 ("Four different types of word vectors are used in this thesis: Pre-trained vectors from Google (word2vec)[,] Pre-trained vectors from Stanford University (GloVe)[,] Vectors trained on the data[,] [and] Random Vectors").

2. Overview of Pendar

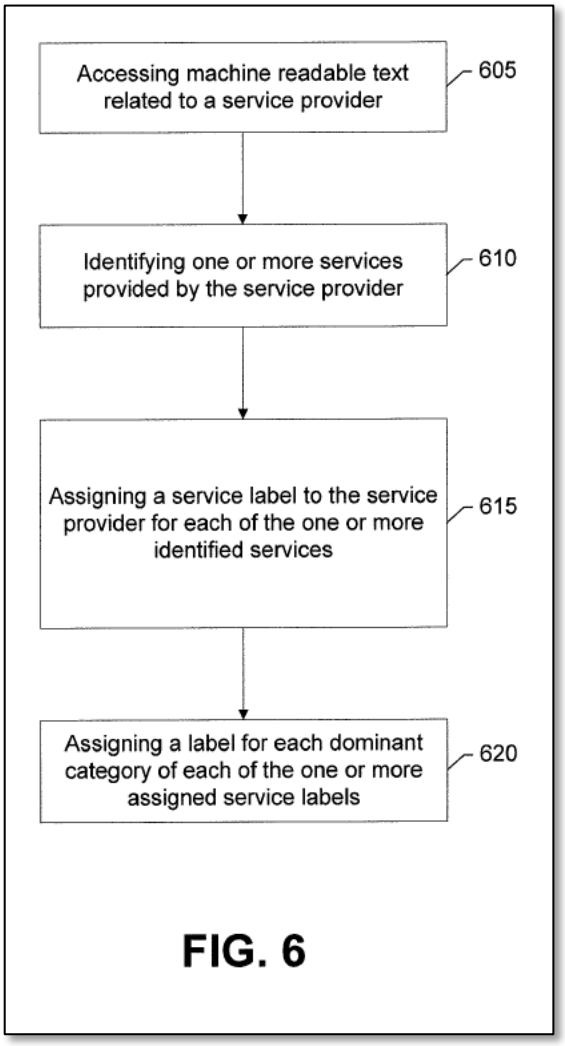
87. Pendar teaches a system and apparatus for classifying and tagging textual data. Pendar (EX1005), 7:58-61 ("Methods, apparatus and computer program products described herein are operable for the classification and tagging of textual data using automatically learned or generated queries"). Specifically, Pendar's "classification and tagging apparatus 100, may be configured for accessing machine-readable text related to a service provider." *Id,* 15:33-35. After accessing the textual data, Pendar's apparatus may identify "one or more services provided by the service provider[.]" *Id.,* 15:39-41. Based on this identification, Pendar's apparatus may then assign "one or more labels to the service provider." *Id.*, 15:46-48. Labels can be assigned by Pendar's apparatus scoring a document and comparing that score to a predetermined threshold. Pendar states:

> As shown in block 515 of FIG. 5, an apparatus, such as a classification and tagging apparatus 100, may be configured for scoring each document in accordance with each of one or more labels. One method of scoring may include summing each of the zero or more weights associated with positive queries. For example, if a first service having a score of "0.25" returns a positive result for a document, and four queries of a second service, having scores of "0.4", "0.5", "0.6", and "0.7" return positive results for the document, **the document may be assigned a score of "0.25" for the first service and "2.2" for the second service.**
>
> As shown in block 520 of FIG. 5, an apparatus, such as a classification and tagging apparatus 100, **may be configured for checking each of the scores, for each of the documents, to a threshold score.** For example, continuing from the previous paragraph where a document is found to have a "0.25" score for the first service and a "2.2" score for the second service, each of the scores may be applied to a threshold, for example "2.0". **Here, the second service may be assigned to the document because the 2.2 score exceeds a predetermined threshold of 2.0.**

Id., 14:37-57 (describing documents being scored, a comparison of document score to a label's threshold value, and assignment of the label to the document if the score exceeds the threshold).

88. Pendar teaches that the labels may be related to one another based on a hierarchy. *Id.*, 15:52-53 ("In another embodiment, one or more labels may be determined to be related to another label in the hierarchy"). For example, Pendar discusses that the label "'burgers' may be associated to 'traditional American cuisine' and 'meat'," so the hierarchical structure of the service taxonomy "includes

at least [three] levels." *Id.,*15:53-57. Moreover, as the labels may be related to one another in a hierarchy, Pendar's apparatus may be configured to assign "a label for each dominant category of each of the one or more assigned service labels." *Id.,* 15:63-66, Fig. 6.



Id., Fig. 6.

3. Claim 1(Pre): A computer-implemented system for improving classification of criminal activities arising from cyber environments, comprising:

89. To the extent the preamble is limiting, Deliu teaches a computer-implemented system that compares "the classification performance of different machine learning algorithms on data from hacker forums." Deliu (EX1004), 65-66. Deliu's machine learning algorithms classify textual data from hacker forums, used by cyber criminals to exhange cyber attack information. *Id.*, 18 ("Different communication channels such as online forums are being used by **cyber criminals** to exchange these services"); *see also id.*, 17 (describing analyzing "Dark Web" forum posts from "[c]yber **criminals** (also called hackers)"). Deliu's machine learning techniques "improve" upon manual analysis, automating or semi-automating the analysis "to enrich [] cyber security controls." *Id.*, 18-19.

90. **In sum**, Deliu teaches "[a] computer-implemented system for improving classification [Deliu's machine learning techniques improving upon previous techniques] of criminal activities arising from cyber environments [Deliu's Dark Web forum posts used by criminals]."

4. Claim 1(a): a processor configured to:

91. Deliu teaches that "[t]he computer system used for running the experiments" uses an "Intel Quad Core i7-3820 3.60GHz" processor. Deliu (EX1004), 73. Deliu's processor is "configured to" execute its experiments via tools

and software, including Python 3.6.0, Sklearn 0.18.1, Theano 0.8.2, and MySQL Community Server (GPL) 5.7.17-log. *Id.*, 73.

92. **In sum**, Deliu teaches "a processor [*Deliu*'s Intel Quad Core-i7-3820] configured to [*Deliu*'s tools and software]" to execute the limitations of Claim 1.

5. Claim 1(a)(1): access a ground truth dataset generated from deep web forum information, the ground truth dataset defining a predetermined tag hierarchy including predetermined tag labels corresponding[2] topics of the deep web forum information;

93. As an initial matter, I note that this claim does not require actually crawling the dark web, only accessing a dataset that was "generated from the deep web." A POSITA would have known that the dark/deep Web crawlers have existed to extract information from these forums. *See* §V.F above. I also note that only claim 11 claims a "crawler" element. *See* §XII below (Ground 7).

94. Deliu teaches accessing a multi-class dataset from the Nulled.IO hacker forum (*i.e.*, *ground truth dataset generated from deep web forum information*) comprising labeled posts (*i.e.*, *including predetermined tag labels*). Deliu (EX1004), 62-64. These labels correspond to topics in the hacker forum (*i.e.*,

[2] The claim language appears to have omitted the word "to." I understand the claim to properly read "predetermined tag labels corresponding [to] topics…].

corresponding [to] topics of the deep web forum information"), such as DDOS, Trojans, SQL Injections, etc. *Id.*; *see also id.*, 76 (Fig. 12).

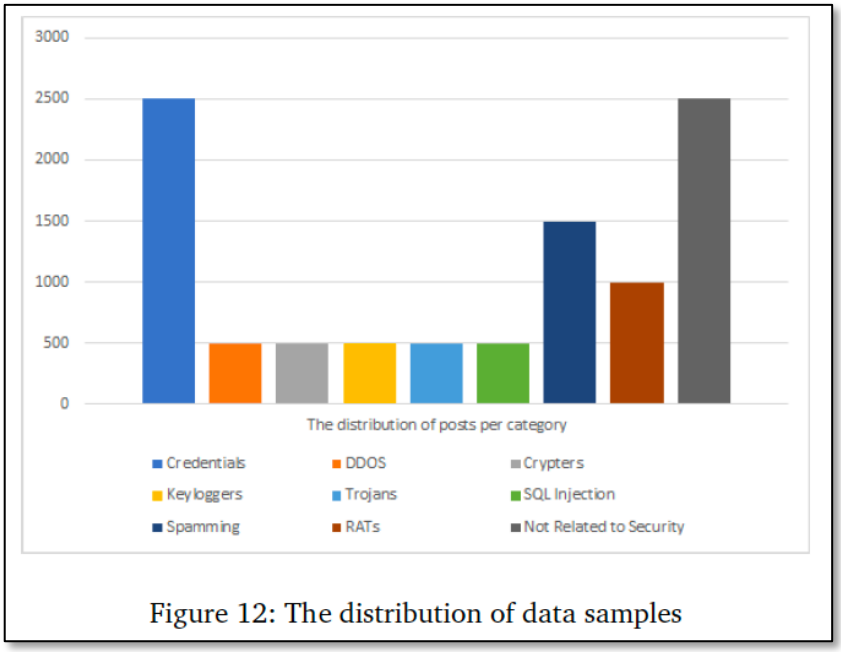


Figure 12: The distribution of data samples

Id., Figure 12.

95. Specifically, Deliu tested its machine learning classification methods "on **a real hacker forum.**" *Id.*, 7, 19, 20; *see also id.,* Title ("Extracting Cyber Threat Intelligence From Hacker Forums"). Namely, Deliu tests its experiments on a leaked dataset generated from the Nulled.IO dark-web forum. *Id.*, 62, 73-74.

96. A POSITA would have understood that Nulled.IO is a deep web forum. For example, the Nulled.IO forum was originally part of the dark web. Deliu (EX1004), 17-18 ("In this thesis, we focus on the last source, Dark Web … we study hacker forums as they represent a considerable part of Dark Web"). Further, Nulled.IO was a well-known Dark Web forum. Cyble (EX1013), 2 ("Nulled is one

of the most notorious dark web forums."). Deliu obtained the forum information from a whistleblower website (http://leakforums.net/thread-719337). Deliu (EX1004), 74.

97. Moreover, the dark web is a subset of the deep web. Cyble (EX1013), 2 ("The dark web, a subset of the deep web, is where anonymity and secrecy intensify. It is intentionally obscured from search engines and requires special software like Tor (The Onion Router) to access it."); Spiceworks (EX1014), 3 ("The dark web is a subsegment of the deep web that can only be accessed with special tools, such as the Tor browser."); *see also* '900 Patent (EX1001), 1:63-64 ("The dark web forms a small part of the deep web that is not indexed by web search engines."). Thus, the Nulled.IO data used by Deliu was originally a dark web forum, and the dark web is a subset of the deep web. Meaning, a POSITA would have understood Deliu's Nulled.IO dataset is "generated from deep web forum information."

98. Like the '900 Patent, Deliu manually labels its ground truth dataset with "predetermined tag labels[.]" *Compare* '900 Patent (EX1001), 3:61-62 ("The ground truth to build a machine learning model **may be hand labeled by field experts**") *with* Deliu (EX1004), 62 ("In order to build a model which is able to filter out posts which are not relevant to security, the presence of labeled samples is required … we constructed a dataset by manually labeling some of the posts in the identified forum").

99. Indeed, Deliu teaches "construct[ing] a multi-class dataset" from the Nulled.IO forum and manually labeling posts with a variety of predetermined tag labels. Deliu (EX1004), 74-76. The distribution of labeled samples per category is shown in Figure 12 below:

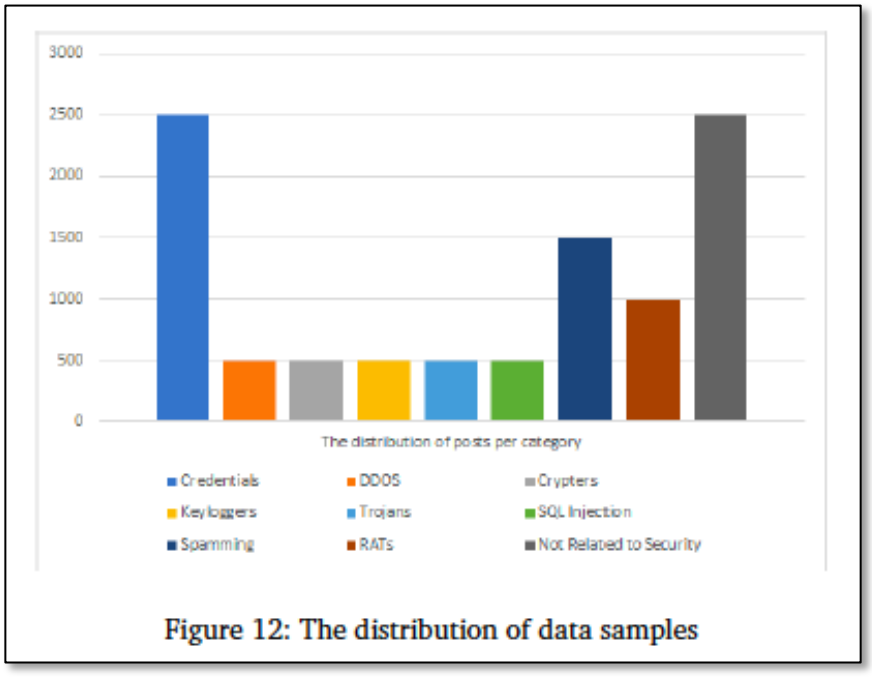


Figure 12: The distribution of data samples

Id., 76 (Fig. 12).

100. These labels *correspond to topics of the deep web forum information*. Deliu's Table 7 below provides illustrative dark web forum posts for each of the categories in the "multi-class dataset":

Table 7: Examples posts from multi-class dataset

Category	Example
Not Related to Security	Hello dear nulled.io community. This is a simple question, what are your favourite movies? ; ; Mines? Idk. Probably Jackie chan movies and/or taken series
Credentials	Hello Guys, just cracked these spotify accounts. It's premium. ; Enjoy. ; ; ; [hide] seand.bertran@gmail.com:Daniel81588 niru660@gmail.com:airbusa380 pskinner63@gmail.com:este11a ellis.nathan@gmail.com:unit24 jano761012@gmail.com:324657 sammy.pierce@gmail.com:kre8tv12 zelus.et.radix@gmail.com:ocelote csoto1251@gmail.com:blah123123 jcarrara24@gmail.com:maryjane yourwifealex@gmail.com:tita2165 ; [/hide] ;
DDOS attack	If i want to ddos someone, which Port should i use? and which method is best? UDP? ; 80 UDP is ok .
Keyloggers	Source code of a simple, litle Keylogger for Windows developed in C++ for Windows only. A good starting point to devlope a private FUD keylogger. ; ; Reply and upvote to unlock the source code link: it
Crypters	Hey guys! ; Here is the cracked Codelux version 3.6.6 It is a crypter to make files undetected by AV. Deobfuscated by li0nsar3c00l, cracked by Meth ; Here is the product description : ; ; ; Download ; [hide]https://mega.co.nz/#!K1wjjLDJ!YuTqbnlPVhjo6ivToCKX6G1G3CdyLfE3FuYmua1J6w[/hide]
Trojans	Enjoy ; ; http://www12.zippyshare.com/v/5AQ1WVsB/file.html TROJAN DETECTED!
SQL Injection	20 SQL injectable's. ; [hide] http//allaces.ru/p/episode.php?id=1' http//www.keswick2barrow.co.uk/faqs.asp?ID=1' http//www.alicantegolf.org/principal.php?idp=1' http//www.bulsu.edu.ph/news.php?id=999999.9 union all select 1,2,3,4,5,6,7777777,8,9,10 http//www.wollin.de/w3.php?nodeId=3'3 http//www.abramat.org.br/site/lista.php?secao=999999.9 union all select 1 ; ; [/hide] ; Have fun!
Spamming	User has 4 posts and spammed VIP section with 3 different fake scripts. Asking for a ban.
RATs	I think, personally, that this RAT is one of the best for its compatibility with all windows machines. ; ; Download Here: [hide] https://megabitload.com/download/index/96758211/ [/hide]

Id., 77 (Table 7).

101. Deliu "accesses" the multi-class dataset to train and test its classifier algorithms. *Id.*, 80-86. Deliu "measure[s] the performance of traditional classifiers … by averaging the results of running 10-fold cross validation for 10 times" for the binary dataset such that in each iteration, "a random sample of approximately 10% is hidden from the training algorithms and used as test data." *Id.*, 80. This same

process is used for the multi-class dataset. *Id*., 85 ("We run all of the experiments reported so far in the multi-class dataset as well").

102. Deliu does not specifically discuss that its multi-class dataset includes hierarchical data such that it defines a "predetermined tag ***hierarchy***[.]" Pendar teaches this.

103. Pendar teaches a corpus-database of prelabeled documents where the labels are taken from a defined hierarchy. Pendar (EX1005), 7:61-64 ("a training method … may be configured to access a corpus that is comprised of annotated, labeled, tagged, or otherwise classified documents or textual portions to generate a set of features"). The "corpus may comprise one or more documents with an associated label such as category (*e.g.*, food and drink) and/or service (*e.g.*, pizza) taken from **a defined service hierarchy or service taxonomy**." *Id.*, 8:30-36. The service taxonomy provides a "**predefined hierarchical structure** that includes at least 3 levels." *Id.*, 15:55-57. In an example, the label "burgers" is associated with "traditional American cuisine" which is associated with "food and drink[.]" *Id.*, 15:46-62.

104. Pendar's service taxonomy is described throughout its disclosure:

> The corpus may comprise one or more documents with an associated label such as category (*e.g.*, **food and drink**) and/or service (*e.g.*, **pizza**) taken from a defined service hierarchy or service taxonomy.

Id., 8:32-36.

> In one example embodiment, where the machine readable text is related to a webpage from an unknown service provider, the service provider may be stored with each of one or more labels corresponding to each of the labels assigned to one or more of the documents that are related to that service provider. For example, the machine readable text may include one or more documents crawled from a website, for example, http://MikeTheLocalServiceProvider.com and two of the documents may part be an extensive menu, where one document focused on Italian cuisine while the other focused on desserts. The first document may be assigned labels such as "**pizza**" and "**spaghetti**" and the second may be assigned labels such as "dessert" and "coffee". As such, MikeTheLocalServiceProvider may be determined as providing "pizza", "spaghetti", "dessert" and "coffee" services. In some example embodiments, the website address or a service provider name alone may be suggestive of a particular label.

Id., 15:1-23.

> As shown in block 615 of FIG. 6, an apparatus, such as a classification and tagging apparatus 100, may be configured for assigning one or more labels to the service provider. As described herein, a label may be associated with a category or sub-category, such as for example, the label "burgers" may be associated with a category of "**food and drink**" from a service taxonomy. In another embodiment, one or more labels may be determined to be related to another label in the hierarchy. For example, the label "**burgers**" may be associated to "**traditional American cuisine**" and "meat". In another embodiment, a service taxonomy may provide a predefined hierarchical structure that includes at least 3 levels. In such cases, a label may be associated with one or more labels from a first category and the one or more labels from the first category are then associated to a label from a second category. For example, the label "**lobster**" may be

> associated to "**seafood**". "Seafood" may then be associated to "**food and drink**".

Id., 15:46-62.

105. Pendar's service taxonomy is depicted in the illustration below, which I have created from its disclosure:

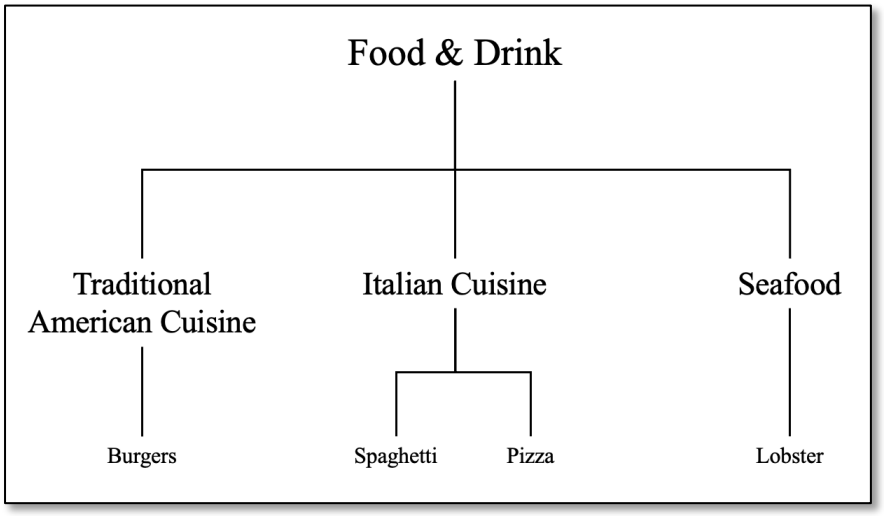


Illustration based on Pendar (EX1005), 8:32-36, 15:1-23, 15:46-62.

106. **In sum**, the Deliu-Pendar combination teaches "access[ing] a ground truth dataset [Deliu's multi-class dataset] generated from deep web forum information [Deliu's dataset generated from Nulled.IO], the ground truth dataset defining a predetermined tag hierarchy [Pendar's hierarchical taxonomy] including predetermined tag labels corresponding [to] topics of the deep web forum information [Deliu's predetermined categories corresponding to cyber threats]."

a. Motivation to Combine Deliu with Pendar (Label Hierarchies)

107. A POSITA would have been motivated to hierarchically label Deliu's multi-class dataset as per Pendar to improve Deliu's classification techniques for at least two reasons. For example, as discussed immediately below, Deliu's SVM, k-NN, and Decision Tree classifiers suffered from a known, "flat" representation of otherwise complex data, where hierarchical classification naturally captures real-world systems. Pendar teaches hierarchical machine learning classification for culinary labels. I note here that hierarchical classification is equally applicable across all fields, whether in the culinary context or in the security context. Thus, a POSITA would have understood that Pendar's culinary labels are applicable and relevant to the deepweb forum classification techniques described and claimed in the '900 Patent.

108. **First**, Deliu expressly envisions utilizing other deep learning algorithms to compare to the algorithms tested in its experiments. Deliu (EX1004), 108 ("Future Work … First, the performance of other deep learning algorithms should be compared with the ones used in this thesis…So, in addition to testing other models of feed-forward neural networks, the utility of other architectures (such as recurrent or recursive deep neural networks) should be explored and studied.").

109. One such improved deep learning algorithm is a hierarchical SVM. As described by Hoffman, an SVM is highly competitive, but it has obvious drawbacks.

Hoffman (EX1015), 1 ("Extensive experimental comparisons […] have evidenced that among the methods available today, SVMs are highly competitive in their classification accuracy and can therefore be considered the state-of-the-art in document categorization. A potential drawback of all of the above mentioned classification methods is that they treat the category structure as 'flat' and that they do not consider relationships between categories, which are commonly expressed in concept hierarchies or taxonomies."). Hierarchical SVMs were known in the art and were known to provide improved data classification. *Id.*, 9 ("We observe that hierarchical SVM outperforms flat SVM in terms of Δ-loss in all cases.").

110. Intuitively, Hierarchical SVM maintains the natural hierarchy used in other well-known classification systems. *Id.*, 1 ("This is witnessed by the fact that virtually all real world classification systems have complex hierarchical structure. This includes traditional systems like Dewey or Library of Congress subject headings, the International Patent Classification (IPC) scheme [21], the Medical Subject Headings (MeSH) maintained by NIH, as well as Web catalogs created by Yahoo!, the Open Directory Project (DMOZ) or LookSmart, to name some of the most important ones."). Thus, a POSITA would have recognized that implementing a hierarchy or taxonomy among the labels would lead to increased performance of the Support Vector Machine, including more accurate and representative real-world classification systems.

111. **Second**, a POSITA would have recognized that Deliu's other classification techniques suffer from the same "flat" representation of otherwise complex relationships. *Id.*, 1 ("The wide range of methods applied to this problem include nearest neighbor classifiers [23], neural networks [19], generative probabilistic classifiers [6, 7], and – more recently – boosting [12] and Support Vector Machines (SVMs) [5], to name just a few … A potential drawback of all of the above mentioned classification methods is that they treat the category structure as 'flat'[.]"). Thus, Deliu's k-NN and Decision Trees classifiers would similarly be improved by implementing a hierarchy or taxonomy among the labels.

112. A POSITA would have had a reasonable expectation of success modifying Deliu to include hierarchical labels as taught by Pendar. Deliu already pre-labels posts into one of several different classes/labels. Deliu (EX1004), 74-76 (discussing the categories shown below in Figure 12).

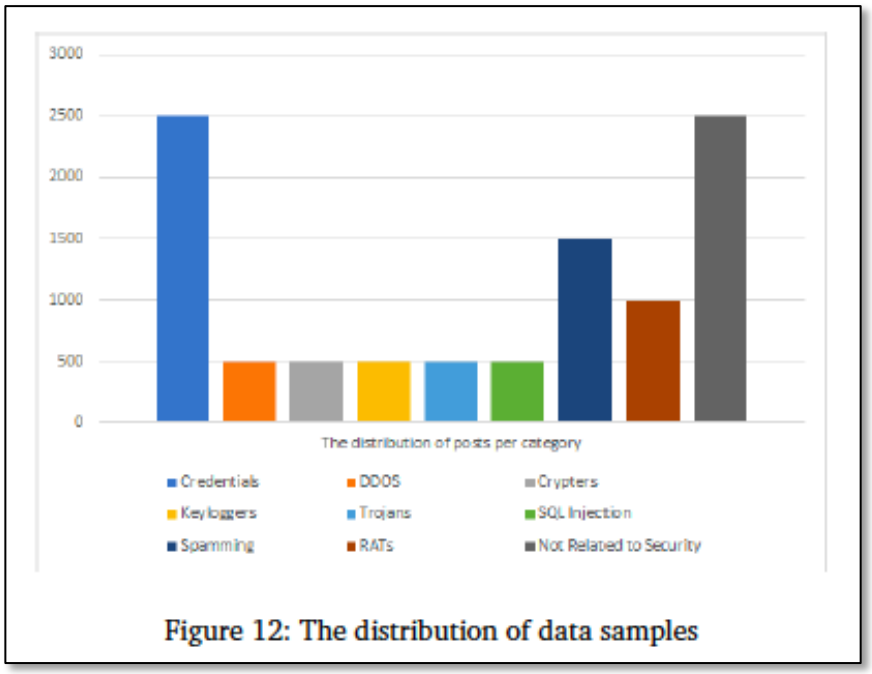


Figure 12: The distribution of data samples

Id., Fig. 12.

113. It would have been well within the skill of a POSITA to pre-label posts hierarchically. For example, data is known to follow a natural hierarchy, so further labeling posts according to this hierarchy would have been a simple modification to Deliu. Hoffman (EX1015), 1 ("A potential drawback of all of the above mentioned classification methods is that they treat the category structure as 'flat' and that they do not consider relationships between categories, which are commonly expressed in concept hierarchies or taxonomies. Such structures, however, are the preferred way in which concepts, subject headings, or categories are arranged in practice."). Classifying ground truth data into a hierarchical form was a well-known strategy in machine learning. *See* §V.A above. Thus, a POSITA would have had success implementing that known strategy in Deliu's methodology.

6. Claim 1(a)(2): access data related to a discussion topic for classification from a deep web forum;

114. Deliu teaches accessing data related to a discussion topic for classification from the deep web forum, Nulled.IO. *See* §VI.A.5 above, Claim 1(a)(1). The Nulled.IO forum is "organized into **topics**, where each topic can have hundred[s] or thousands of posts." Deliu (EX1004), 104. "The main topics related to security in the Nulled.IO forum are … leaked credentials, web and network attacks (SQL injection, DDOS), spamming, and different malware types (Trojans, keyloggers, crypters, RATs, etc)[.]" *Id*., 105; *see also id.*, 119-121 (Appendix A2 providing tables of "[t]opics of posts from Nulled.IO forum"). Using "Classification Methods[,]" Deliu classifies forum posts into categories. *Id.*, 74-76, Fig. 12.

115. **In sum**, Deliu teaches "access[ing] data related to a discussion topic for classification [accessing Deliu's Nulled.IO forum data that is organized into topics] from a deep web forum [Nulled.IO is from the dark web, a subset of the deep web]."

7. Claim 1(a)(3): extract a set of features from the data by assigning word vectors or paragraph vectors to the data; and

116. Deliu teaches extracting features from the Nulled.IO dataset using "Bag-of-Words" and "n-gram" feature extraction techniques. Deliu (EX1004), 65-66, 85. These features are shown in annotated Table 19 below:

Features	k-NN	Decision Trees	SVM
word (uni+bi)-grams	37.48	96.41	**96.93**
character trigrams	68.07	95.96	**98.62**
character(bi+tri)- grams	81.36	95.98	**98.59**
bag-of-words	66.76	96.45	**97.27**

Table 19: Classification accuracy of three classifiers on the multi-class dataset: k-Nearest Neighbors, Decision Trees, and Support Vector Machines

Id., 86 (Table 19, annotated); *see also id.*, 87 (Table 21).

117. Deliu teaches that both bag-of-words and n-gram assign word vectors. *Id.*, 65-66. Bag-of-words is "one of the most common methods for text classification," outputting feature vectors representing "word frequency within each document[.]" *Id.*, 65. "That is, the **dimensionality of the feature vectors** is equal to the number of unique words in all documents in the dataset." *Id.* Deliu explains:

> For illustration, let the dataset consist of three documents: S1="The attacker exploited a vulnerability on the server", S2="The server was quickly compromised", and S3="It was a Trojan virus". It total, there are **13 unique words, which represents the size of the feature vectors**. Should the value of features be the raw frequency of the words then the **feature vectors for the three documents will be:** $\mathbf{S_1}$ = [1, 1, 0, 1, 0, 1, 0, 1, 2, 0, 0, 1, 0], $\mathbf{S_2}$ = [0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1], and $\mathbf{S_3}$ = [1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1], where $S_n = [v_1, v_2, ...v_{13}]$ and words are alphabetically sorted. For binary values, all non-zero features will be changed to 1, whereas these values are divided by their corresponding inverse document frequencies for TF-IDF features.

Id., 65.

118. In other words, a bag-of-words extracts all unique words across all of the documents, then assigns a vector representation to each document based on the occurrence of words within the document. *See* §V.B.1 above. N-gram similarly assigns word vectors but considers "word order[.]" *Id*., 66 ("word order is considered in a local context by n-gram features, which are defined as a sequence[] of *n* consecutive words/characters"). N-grams extracts all unique n-gram features from a document, that is, word or character sequences of length 1, 2, … n. Then, similar to bag-of-words, it assigns a vector representation to each document based on the occurrence of n-grams within the document. *See* §V.B.2 above.

119. **In sum**, Deliu teaches "extract[ing] a set of features from the data by assigning word vectors [Deliu assigning bag-of-words or n-gram feature vectors] or paragraph vectors to the data[.]"

8. Claim 1(a)(4): apply a machine classifier to the set of features to generate a prediction list of tags for classifying the discussion topic, wherein the prediction list includes a prediction probability value for each tag of the plurality of tags; and

120. Deliu applies machine classifiers, such as SVM, k-NN, Decision Trees, and CNNs, to the extracted features to generate labels for each discussion post in the Nulled.IO forum. Deliu (EX1004), 78-87 ("6.3 Classification of forums posts"); *see also id.*, 76 (Figure 12 illustrating distribution of labeled posts). Deliu "measure[s]

the performance of traditional classifiers" in its datasets using different features. *Id*., 80. This is summarized in *Deliu*'s Table 19 below:

Features	k-NN	Decision Trees	SVM
word (uni+bi)-grams	37.48	96.41	**96.93**
character trigrams	68.07	95.96	**98.62**
character(bi+tri)- grams	81.36	95.98	**98.59**
bag-of-words	66.76	96.45	**97.27**

Table 19: Classification accuracy of three classifiers on the multi-class dataset: k-Nearest Neighbors, Decision Trees, and Support Vector Machines

Id., 86 (Table 19).

121. Deliu discusses the relationships between classifiers and feature vectors. *Id.*, 87 (Table 21 comparing results of different "Classifier[s]" and "Features"); *see also id.*, 35 ("Most of the existing machine learning algorithms are able to build predictive models or infer intrinsic structural information **only when the data is presented in the form of feature vectors**."), 41 ("Since machine learning algorithms are able to build predictive models only **from the features extracted** from raw data, their success is closely related to the quality of the features").

122. Deliu applies machine classifiers to the set of features to generate tags for classifying the discussion topic. Indeed, each of the Features identified in Deliu's Table 19, reproduced above, (*e.g.*, word (uni + bi) grams, character trigrams, character (bi + tri) grams, and bag-of-words) assigns feature vectors to documents.

See §VI.A.7 above, Claim 1(a)(3). The classifiers identified in Deliu's Table 19 (*e.g.*, k-NN, Decision Trees, and SVM) and Table 21 (*e.g.*, Convolution-NN) use the extracted features as input. Deliu (EX1004), 35, 41. The classifiers output predicted labels for the document. For example, Deliu generates predicted labels for nine different categories of forum posts relevant to cyber security. *Id.*, 74-76, Fig. 12; *see also* Deliu Paper (EX1016), 3 ("The multinomial data set has a single class/label for the data from irrelevant posts and then several different classes/labels for the relevant posts").

123. Deliu does not expressly discuss generating "*a prediction list of tags … wherein the prediction list includes a prediction probability value for each tag of the plurality of tags*[.]" Pendar teaches this.

124. Pendar teaches a "tagging apparatus" that "assigns labels" by executing queries on documents with machine readable text. Pendar (EX1005), 14:19-27, 15:27-62, Fig. 5. "[E]ach set of queries [is] associated with a label" such that multiple labels can be applied to a document. *Id*., 14:25-28, 14:34-36. To apply "one or more labels[,]" Pendar calculates a prediction score for each query. *Id*., 14:37-47. "One method of scoring may include summing each of the zero or more weights associated with positive queries." *Id*., 14:40-42. Pendar compares these scores to a "threshold" and "for each score meeting the threshold score, [Pendar] assign[s] one or more labels associated with the score to the document." *Id.*, 14:58-

67. "[I]f a document retrieved from an electronic source has summed scores exceeding a pre-defined threshold for 'pizza' and 'spaghetti', each label may be associated with the document[.]" *Id*.; *see also id*., 15:27-16:3.

125. Pendar predicts multiple labels for a document (*i.e.*, "*generate a prediction list of tags for classifying the discussion topic*"). Specifically, Pendar teaches that up to *n* labels can be predicted. *Id.*, 14:28-36 ("For example, each query from a set of queries with an associated label of a first service, a second service and an n^{th} service may be applied.").

126. Pendar's prediction list includes *a prediction probability value for each tag of the plurality of tags*. *Id.*, 14:37-67 ("One method of scoring may include summing each of the zero or more weights associated with positive queries. For example, if a first service having a score of '0.25' returns a positive result for a document, and four queries of a second service, having scores of '0.4', '0.5', '0.6', and '0.7' return positive results for the document, the document may be assigned a score of '0.25' for the first service and '2.2' for the second service."). The weights/scores are a value between 0 and 1, designed "to enable classification of new documents." *Id.*, 9:49-53. *See also id*., 9:56-62. In particular, the score is "indicative of the informativeness of the query" as it is based on the precision rate and/or the recall rate of the query. *Id.*, 12:44-64 ("For each of the queries **meeting or exceeding a precision threshold**, the score may be calculated and **the respective**

precision and/or recall rate may be factored into the calculation of that score."). Thus, the "summed" "scores" for each label are "values" that are compared to a threshold to predict a label and are therefore *prediction probability* values as claimed.

127. **In sum**, the Deliu-Pendar combination teaches "apply[ing] a machine classifier [Deliu's SVM, k-NN, Decision Tree, CNN] to the set of features to generate a prediction list [Pendar's hierarchical list] of tags [Deliu's labels] for classifying the discussion topic, wherein the prediction list includes a prediction probability value [Pendar's summed score] for each tag of the plurality of tags."

a. Motivation to Combine Deliu with Pendar (Multiple Labels)

128. As described in §VI.A.5.a above for Claim 1(a)(1), hierarchical classification is equally applicable across all fields, whether in the culinary context or in the security context. Thus, a POSITA would have understood that Pendar's culinary labels are applicable and relevant to the deepweb forum classification techniques described and claimed in the '900 Patent. Therefore, a POSITA would have been motivated to modify Deliu's classifiers to implement Pendar's multi-label querying techniques to facilitate assigning more than one label to each document, for at least the following two reasons.

129. **First,** Deliu explains that categories selected for his study were "simple enough" for effective classification, but acknowledged that "more complex feature

space topologies" exist. Deliu (EX1004), 103 ("The results of the experiments show that the traditional SVM classifier with character n-gram features perform at least as good as the ConvNN. […] We believe that the reason for this is the number of samples in the dataset; this was also stated by Zhang *et al*. [68] who proposed a different deep ConvNN and stated that its performance starts to get better than traditional approaches only when the dataset contains millions of samples."); *see also* Deliu Paper (EX1016), 7-8 ("[T]he boundaries between the different classes in the feature space are simple enough for the SVMs to easily separate the classes. In the case of more complex feature space topologies that do not lend themselves to easy separation via SVM, then we would expect the CNN performance to exceed the SVM.").

130. Complex topologies would benefit from classifying a document into multiple categories to account for overlap where an item can belong to several categories at once, such as in Pendar's exemplary hierarchical taxonomy. A POSITA would have understood that being able to encode more complex topologies would improve the classification accuracy. Multiple, hierarchical categories provide a more realistic classification for the dataset. Hoffman (EX1015), 1 ("This is witnessed by the fact that virtually all real world classification systems have complex hierarchical structure."). Additionally, hierarchical categories account for ambiguity

in the data by ascribing one or more dominant labels to a document to provide more thorough context information about the document.

131. **Second,** Pendar's predicted labels and scores would have provided important information regarding the "informativeness" of each label. Pendar (EX1005), 12:44-49, Fig. 3. A POSITA would have understood that this data would have enhanced Deliu's classifiers to more effectively differentiate and classify documents in more complicated feature space topologies, as contemplated by Deliu. Deliu's feature space could be made more granular by additionally classifying the software or platform for which a specific threat is targeted. For example, classifying zero-day exploits according to their targeted platform, as seen in Deliu's Table 32 below:

Table 32: Examples of posts sharing 0days

Description	Post
OS X 0day	[hide]http://pastebin.com/i9KSpnRb[/hide] ; Not sure how this works lol ; "OS X 0day - works on latest version as of 4/30/15 BO exploitation @ fontd, allows payload to run code with fontd privileges." ; It's C syntax
OpenSSH 0day	/* * * Priv8! Priv8! Priv8! Priv8! Priv8! Priv8! Priv8! * * OpenSSH <= 5.3 remote root 0day exploit (32-bit x86) * Priv8! Priv8! Priv8! Priv8! Priv8! Priv8! Priv8! * * */ void usage(char argv) ; ; printf(" HATSUNEMIKU"); ; ; printf("OpenSSH <= 5.3p1 remote root 0day exploit"); ; ; printf(By: n3xus"); ; ; printf("Greetz to hackforums.net"); ; ; printf("Keep this 0day priv8!"); ; ; printf("usage: ; ; exit(1); ...
.doc 0day	Hello everybody... Me and my friend recoded and refuded 0day silent .doc exploit. We plan to sell it and ask you guys what do you think, ;how much this is worth and what should be the price for this exploit?https://www.youtube.com/watch?v=IqILJ2gfkgM ; Scan output 0/35:;scan.majyx.net/scans/result/b37c6e8be34752c3cfed82c27edcf927b85ce6b2
Joomla 0day	Hi everyone, Today , I'll share a private flaw, which is no longer seen as 2 person is already shared on the net it has paid to the deep 6months ago it is a Joomla based 0day hd flv player plugin so the flaw allows the recovery of the config file and with her you can exploit the DB seen that there are in the password; [hide]Please type in Google you inurl: /component / hdflvplayer / or inurl: com_hdflvplayer the feat you take a URL that contains the flaw. and paste the text you like thatlesitequicontienlafaille//components/com_hdflvplayer/hdflvplayer/download.php?f=../../../configuration.php /components/com_hdflvplayer/hdflvplayer/-download.php?f=../../../configuration.php ; Please look for PHPMYADMIN only by analyzing the site etc, for example if it is an OVH website you go to google you tapper phpmyadmin OVH and you put the logs that you download via the flaw.
Firefox 0day	. [hide] http://0day.today/exploit/24128[/hide]decent imo. ;this metasploit module gains remote code execution on firefox 35-36 by abusing a privilege escalation bug in resource:// uris. pdf.js is used to exploit the bug. this exploit requires the user to click anywhere on the page to trigger the vulnerability

Deliu (EX1004), 96 (Table 32).

132. A POSITA would have had a reasonable expectation of success modifying Deliu's classifiers to include Pendar's similar classification techniques. Pendar teaches "learned" queries. Pendar (EX1005), 1:8-10. ("Embodiments of the invention relate, generally, to classification and tagging of textual data with automatically learned queries."). The queries inform whether a document is relevant to a label. *Id.*, 12:44-49 ("For each of the queries meeting or exceeding a precision threshold, the score may be calculated and the respective precision and/or recall rate

may be factored into the calculation of that score."). Following Pendar's teachings, a score for each predicted label is obtained by summing the scores of each positive query. *Id.*, 14:48-57. In the modified system, Deliu's classifiers would run Pendar's "learned" queries for each category for a particular document, obtaining a score for each of *n* predicted labels. *Id.*, 14:34-36. Deliu would then run its traditional classification techniques, including SVM, k-NN, Decision Tree, and CNN, for each predicted label to assign the labels that exceed a threshold. Under each of these classification techniques, a label is assigned when its score exceeds a threshold. *See* §V.C above.

133. Moreover, multiple labels could be generated for a single document. Indeed, multiple labels associated with a single document accommodate hiearachical classifications according to well-known principles. Hoffman (EX1015), 3 ("2.5 Hierarchical Multilabel Classification … The same idea can be carried out in the multilabel case of overlapping binary classes, where document may be assigned to multiple categories."), 1 ("This is witnessed by the fact that virtually all real world classification systems have complex hierarchical structure.").

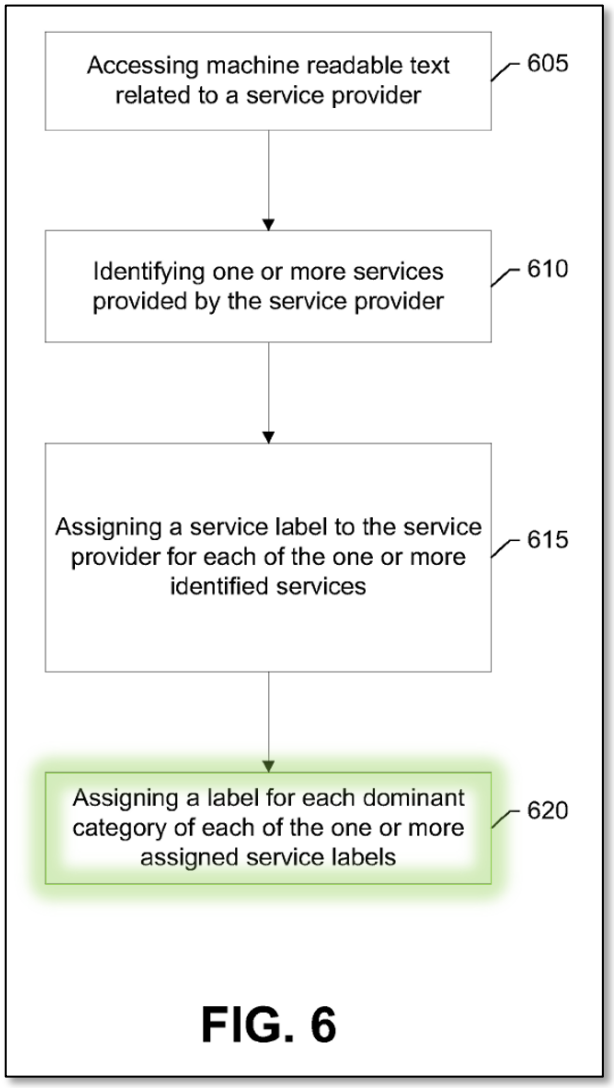
9. Claim 1(a)(5): add all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value.

134. The Deliu-Pendar combination teaches generating multiple predicted tags (*i.e.*, *prediction list*) and summed scores for those tags (*i.e.*, *prediction*

probability value) for posts in the Nulled.IO dataset. *See* §VI.A.8 above, Claim 1(a)(4). Pendar further teaches an additional label assignment technique where "dominant categories" are assigned for each assigned label (*i.e.*, *add all parent tags*). Pendar (EX1005), 15:46-16:3 ("As shown in block 620 of FIG. 6, an apparatus, such as a classification and tagging apparatus 100, may be configured for assigning a label for each dominant category of each of the one or more assigned service labels. Continuing the example from the previous paragraph, in an instance in which a label 'lobster' is determined, the label 'lobster' may be assigned, a dominant label 'seafood' may be assigned, and a dominant label 'food and drink' may be assigned.").

135. Pendar further appreciates that "one or more labels may be determined to be related to another label in the hierarchy." *Id*., 15:52-53. "For example, the label 'burgers' may be associated to 'traditional American cuisine' and 'meat'." *Id*., 15:53-55. This is exemplary of Pendar's "predefined hierarchical structure that includes at least 3 levels". *Id.*, 15:55-57. Pendar teaches in "such cases, a label may be associated with one or more labels from a first category and the one or more labels from the first category are then associated to a label from a second category." *Id*., 15:57-62 ("the label 'lobster' may be associated to 'seafood'. 'Seafood' may then be associated to 'food and drink'").

136. Continuing the example, "in an instance in which a label 'lobster' is determined, the label 'lobster' may be assigned, a dominant label 'seafood' may be assigned, and a dominant label 'food and drink' may be assigned." *Id*., 15:66-16:3. This process is shown in Pendar's annotated Figure 6 below:



Id., Fig. 6 (annotated).

137. Pendar teaches applying all dominant labels (*i.e.*, *parent tags*) for tags that have been assigned a label. A POSITA would have understood that the

dominant/sub-dominant categories, as taught by Pendar, refer to a parent/child taxonomy.

138. In the modified system, Deliu-Pendar assigns labels based on its classifiers comparing scores to thresholds. *See, e.g.*, *id.*, 14:48-51 ("checking" the summed score "to a threshold score"), 14:51-57 (if the summed score "exceeds a predetermined threshold[,]" then the label is applied to the document). Following the label assignment, all dominant category labels are assigned. *Id.*, 15:63-16:3 ("[I]n an instance in which a label 'lobster' is determined, the label 'lobster' may be assigned, a dominant label 'seafood' may be assigned, and a dominant label 'food and drink' may be assigned."). Deliu-Pendar therefore teaches adding all dominant labels associated with a label to the prediction list for labels that exceeded a predetermined threshold (*i.e.*, *add all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value*).

139. **In sum**, the Deliu-Pendar combination teaches "add[ing] all parent tags [Pendar's adding all dominant-level labels] associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value [Pendar checking the summed score against a predetermined threshold for the label].

a. Motivation to Combine Deliu with Pendar (Parent/Child Tags)

140. As described in §VI.A.5.a above for Claim 1(a)(1), hierarchical classification is equally applicable across all fields, whether in the culinary context or in the security context. Thus, a POSITA would have understood that Pendar's culinary labels are applicable and relevant to the deepweb forum classification techniques described and claimed in the '900 Patent. Therefore, for the same reasons stated in §VI.A.8.a above, Claim 1(a)(4), a POSITA would have been motivated to improve Deliu's machine classifier with Pendar's hierarchical parent/child classification techniques, for at least the following two reasons.

141. **First,** Pendar's techniques include adding all labels for each "dominant category" (*i.e.*, *parent tags*) of a classified label. Pendar (EX1005), 15:46-16:3. A known problem with hierarchical classification is inconsistency across all hierarchical levels. One prominent example is omitting a higher-level classification—a false negative prediction. Modifying Deliu with Pendar would help reduce false negative predictions of higher-level tags by predicting the most specific lower-level tag then auto-populating the higher-level tags. In this way, all lower-level tags that are correctly predicted necessarily predict the higher-level classification. This increases classifier performance and reduces missed labels. It also helps reduce semantically inconsistent predictions that contradict the natural hierarchy in the dataset. Using Pendar's example taxonomy, if a sub-dominant label

'lobster' is assigned to a document, then the dominant-level label 'Italian cuisine' is also added. Thus, Pendar maintains the correct hierarchy, even into the dominant-levels, and avoids a false negative prediction of 'Italian cuisine.'

142. **Second,** a POSITA would have found it obvious to modify Deliu's classifier to add all dominant labels of a classified label for accurate and thorough data classification. This proposed modification would be applying a known technique to a known device ready for improvement to yield predictable results. For example, Pendar's method of assigning labels to each dominant category of that label was a technique known in the art. *See* §V.A above. Deliu's machine learning methods were also known in the art. Deliu (EX1004), 7 ("In the first phase, the recent developments in Deep Learning are compared against more traditional methods for text classification."). Deliu also acknowledges areas of improvement and future work. *Id.*, 108 ("Future Work … The research done in this thesis has many potential extensions. First, the performance of other deep learning algorithms should be compared with the ones used in this thesis … the utility of other architectures (such as recurrent or recursive deep neural networks) should be explored and studied."). Finally, as discussed immediately above, this combination would have resulted in improved classifier performance.

143. A POSITA would have had a reasonable expectation of success modifying Deliu to add all dominant labels of a classified label. This is a simple and

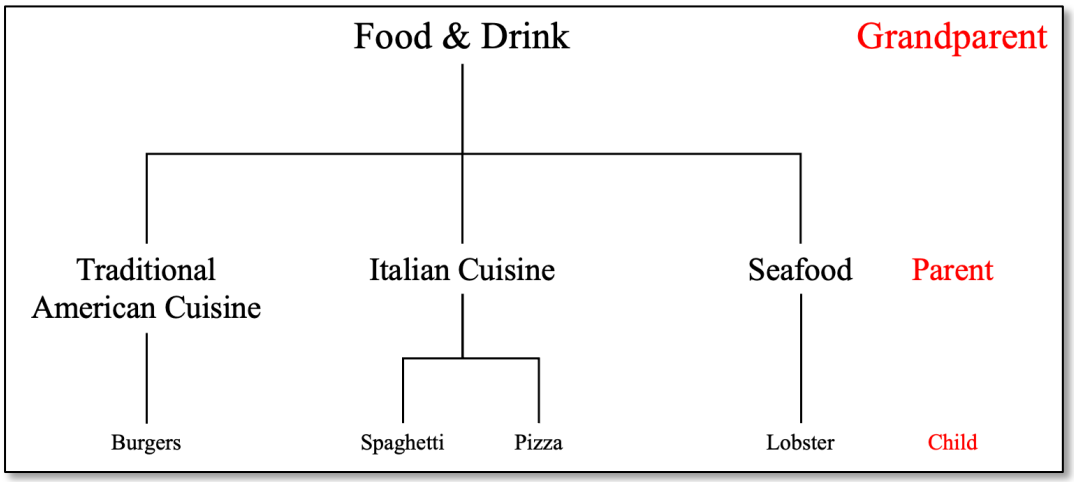
known technique for improving classification that would have leveraged the natural hierarchy in Deliu's datasets, utilizing the same hardware and requiring minor modifications to software that would have been well within the skill of a POSITA at the time. Deliu's hardware already performed the disclosed classification techniques, and a POSITA would have expected the same hardware to perform the modified classification.

144. Further, modifying Deliu's software, which already classified documents according to known machine learning techniques, would have been an insignificant change to further include the steps of adding all dominant-level labels. Maintaining this class hierarchy was known in the art and is taught early on to computer scientists. *See* §V.A above. Moreover, the modification would have predictably ensured that class hierarchy is maintained. *See,* Ramírez-Corona (EX1008), 1 ("In hierarchical classification, an example that belongs to certain class automatically belongs to all its superclasses").

B. Claim 2: The computer-implemented system of claim 1, wherein a predetermined tag hierarchy is divided into parent tags and child tags, wherein a topic represented by a child tag is representative of a subset of a topic represented by a parent tag, wherein each child tag corresponds to a particular parent tag, and each tag is assigned a corresponding probability value.

145. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3–VI.A.9 above, Claim 1.

146. The Deliu-Pendar combination also teaches that its dataset from the Nulled.IO hacker forum includes a predetermined tag hierarchy. *Id*., Claim 1(a)(1). Namely, Pendar teaches a tag hierarchy with at least three levels. Pendar (EX1005), 8:32-36, 15:1-23, 15:46-62. This tag hierarchy is illustrated in a figure I created based on Pendar's disclosures annotated below:



147. A POSITA would have understood that the three hierarchical levels correspond to a parent-child relationship, being divided into parent tags and child tags. These form a classic tree structure; *see* §V.A above. Further, a POSITA would have understood the child tags are a subset of a topic represented by the parent tag. For example, "the label 'lobster' [(*child tag*)] may be associated to 'seafood' [(*parent tag*)]." Pendar (EX1005), 15:60-62.

148. In other examples, the dominant-level parent tag, "Italian cuisine[,]" corresponds with the sub-dominant level child tags "spaghetti" and "pizza." *Id.*, 15:6-15. A POSITA would have understood that "spaghetti" is *a topic represented*

by a child tag and that it is *representative of a subset of a topic represented by a parent tag*—Italian cuisine. For example, as illustrated above, Italian cuisine encompasses at least Spaghetti and Pizza. Thus, each of the tags "spaghetti" and "pizza" correspond to a particular parent tag, Italian cuisine. *See also, id.*, 15:53-56 ("the label 'burgers' may be associated to 'traditional American cuisine' and 'meat'"), 15:55-62 ("[T]he label 'lobster' may be associated to 'seafood'. 'Seafood' may then be associated to 'food and drink'").

149. Finally, for the same reasons discussed in §VI.A.8 above for Claim 1(a)(4), Pendar teaches *each tag is assigned a corresponding probability value*. Pendar (EX1005), 14:34-36 ("an associated label of a first service, a second service and an n^{th} service may be applied"). Specifically, Pendar teaches that labels are applied by summing scores of each query returning a positive result for a document such that when a document receives a summed score above a predetermined threshold, the labels are applied. *Id.*, 9:63-10:7, 14:34-57.

150. Pendar describes that the document is "assigned" the summed score. *See, e.g.,* Pendar (EX1005), 14:42-47 ("the document may be **assigned** a score of '0.25' for the first [label] and '2.2' for the second [label]"). In the case of multiple labels (*e.g.*, up to *n* applied labels) for a single post, values are assigned for each applied label. This includes labels following a natural hierarchy such that each of a child and parent labels are applied to the post with assigned values. For example, a

post may have an assigned value greater than a threshold to apply the label "Trojan" while also having an assigned value greater than a threshold to apply the label "Security." In this case, both the dominant-level label (*e.g.*, "Security") and the sub-dominant level label (*e.g.*, "Trojan") would be applied to the post.

151. **In sum**, the Deliu-Pendar combination teaches "[t]he computer-implemented system of claim 1, wherein a predetermined tag hierarchy is divided into parent tags and child tags [Pendar's hierarchichal levels], wherein a topic represented by a child tag [Pendar's sub-dominant level label, *e.g.*, "Spaghetti"] is representative of a subset of a topic represented by a parent tag [Pendar's dominant level label, *e.g.*, "Italian cuisine"], wherein each child tag corresponds to a particular parent tag [Pendar's sub-dominant label corresponding to a dominant level, *e.g.*, "Spaghetti" corresponds to "Italian cuisine"], and each tag is assigned a corresponding probability value [Pendar's summed score]."

C. Claim 4: The computer-implemented system of claim 1, wherein the processor is configured to add all parent tags of the plurality of tags corresponding to a child tag if a probability value associated with the parent tag is above the first predetermined threshold value.

152. As discussed in §IV.C above, I have adopted Petitioner's interpretation for Claim 4, requesting that the Board correct a typographical error. *See* §IV.C above. The corrected claim language is reproduced below:

Claim 4

> "wherein the processor is configured to add all parent tags of the plurality of tags corresponding to a child tag if a probability value associated with the ~~**parent**~~ **[child]** tag is above the first predetermined threshold value."

153. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1.

154. Claim 4 specifies that the probability value of the child tag must be "***above***" the threshold value. Pendar teaches this. Pendar describes "checking" (*i.e.*, *comparing*) the summed score of a child label "to a threshold score" to determine whether that label should be assigned. *See* §VI.A.9 above Claim 1(a)(5); Pendar (EX1005), 14:48-51. If the summed score "**exceeds** a predetermined threshold," then the label is applied. Pendar (EX1005), 14:51-57; *see also id*., 10:6-7.

155. Pendar further teaches that if a document is assigned a label, then the label for each "dominant category of **that label**" is also assigned. *See* §VI.A.9 above, Claim 1(a)(5); Pendar (EX1005), 15:63-16:3, Fig. 6. In other words, when a document is assigned a sub-dominant label (*i.e.*, *child tag*) because its summed score exceeds a threshold, then the label(s) of the dominant category (*i.e.*, *parent tags*) are also assigned. Pendar (EX1005), 15:63-16:3. The Deliu-Pendar combination therefore teaches adding all dominant labels (parent tags) if a probability value associated with the sub-dominant label (child tag) is above the first predetermined threshold.

156. **In sum**, the Deliu-Pendar combination teaches "[t]he computer-implemented system of claim 1, wherein the processor is configured to add all parent tags [Pendar's dominant-level labels] of the plurality of tags corresponding to a child tag [Pendar's sub-dominant level label] if a probability value [Pendar's summed score] associated with the parent tag is above the first predetermined threshold value."

D. Claim 5: The computer-implemented system of claim 1, wherein the set of features are extracted using textual feature extraction to identify tags associated with topic titles of the data.

157. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1.

158. Deliu teaches extracting features using the textual feature extraction algorithms "Bag-of-Words" and "n-gram[.]" *See* §VI.A.7 above, Claim 1(a)(3) (noting that Bag-of-words is "one of the most common methods for text classification" for outputting word vectors and that N-gram similarly assigns word vectors but also considers "word order"); *see also* Deliu Paper (EX1016), 4 (describing Bag-of-Words as "a common method of **representing text documents as feature vectors**" and describing n-gram features as "sequences of ***n* consecutive words/characters**[.]").

159. As discussed in §VI.A.5 above for Claim 1(a)(1), Deliu describes hacker forum posts labeled as DDOS, Trojans, SQL Injections, etc. Deliu (EX1004),

74-76, Fig. 12. These labels correspond to topics in the forum. *Id.*, 77 (Table 7). Deliu further describes applying Latent Dirichlet Allocation (LDA) to improve the "validation of the labeling process[.]" *Id.*, 91; *see also id.*, 31 (providing background on LDA "topic modeling"). Deliu's Table 26 shows results from validating the SQL injection classification (*i.e.*, *tag*, highlighted in **orange**). The LDA technique identifies topics (*i.e.*, *topic titles*, highlighted in **green**) associated with the SQL injection tag. *Id.*, 91.

Table 26: Topics from SQL injection documents

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
sqli	man	ty	program	explanation	thanks	interested	sentry	php	sqli
sql	thanks	sql	sqli	website	sqli	love	mba	58	dumper
use	injectable	injection	dumper	vulnerability	sql	hits	vulnerabilities	http	hide
dorks	dorks	xss	does	based	injection	lol	idk	www	thanks
dumper	sql	possible	kind	explain	tutorial	hq	mysql	id	need
injection	sqli	error	league	crack	site	sqli	database	select	http
learn	hide	injections	net	help	work	cracking	sqli	91	com
know	working	use	source	understand	hack	gyazo	rdp	union	got
want	urls	site	lol	experienced	learning	dorks	proxies	999999	dorks
good	dumper	products	error	hacking	lot	dumping	errors	com	download

Id., Table 26 (annotated); *see also id.*, 119-121 (showing topic titles associated with the other described tags).

160. **In sum**, the Deliu-Pendar combination teaches "[t]he computer-implemented system of claim 1, wherein the set of features are extracted using textual feature extraction [Deliu's Bag-of-Words and n-gram] to identify tags associated with topic titles of the data [Deliu's topics identified via LDA]."

E. Claim 10: The computer-implemented system of claim 1, wherein the set of features are extracted from the data using a neural network, wherein the neural network is configured to learn a continuous distributed vector representation for documents and words within the data.

161. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1.

162. Deliu teaches testing additional classification techniques on its Nulled.IO dataset, including using "Deep Neural Network" classifiers such as a "Convolutional Neural Network" (CNN). Deliu (EX1004), 82. Deliu's CNN classifier operates materially similar to its traditional approaches, and as modified by Pendar, would teach the limitations of claim 1 for the same reasons discussed with respect to Pendar's other classifiers. Namely, Deliu's CNN would implement Pendar's hierarchical classification techniques, including its queries, in the same material way as Deliu's SVM, k-NN, or Decision Tree Classifiers. It was known in the art that convolutional neural networks could be used for hierarchical data classification. *See, e.g.*, Zhu (EX1017), 1 ("In this paper, we introduce a special CNN based model integrated with the prior knowledge of hierarchical category relations"); *see also e.g.*, Yan (EX1018), 1 ("In this paper, we introduce hierarchical deep CNNs (HD-CNNs) by embedding deep CNNs into a two-level category hierarchy"); *see also* §V.D above.

163. Deliu describes feeding several different vectorized inputs to its CNN, including "paragraph vectors[.]" Deliu (EX1004), 65 ("The following is a brief overview of supervised classification algorithms used[.]"), 66 ("Paragraph Vectors"), 82 (Table 13 "Paragraph Vectors"). Deliu's Paragraph Vectors extract features using a neural network. *Id.*, 66 (discussing Paragraph Vectors follow the same principles as word embeddings). Word embeddings are a "**continuous vector representation** model" and consist "of a simple feed-forward **neural network**[.]" *Id*., 36. Paragraph Vectors parameterize both "the **entire text** (paragraph)" and "the vectors of **context words**" to predict the next word in the paragraph. *Id.*, 66. The '900 Patent similarly discusses PV-DM (Distributed Memory Model of Paragraph Vectors):

> A paragraph vector is a supervised framework that may be implemented by the system that learns **continuous distributed vector representations** for documents instead of words. The text in the documents can be of variable length. As shown in FIG. 4, **every paragraph is mapped to a unique vector**, represented by a column matrix D and **every word is also mapped to a unique vector**, represented by a column in matrix W. The paragraph vector and word vectors are averaged or concatenated to predict the next word in a context. The only difference between this Doc2vec PV-DM model compared to the Word2vec framework is the addition of a paragraph vector along with the word vectors.

'900 Patent (EX1001), 5:1-13.

164. This further comports with the understood meaning of Paragraph Vectors. Le (EX1019), 1 (Paragraph Vector is "an unsupervised framework that learns continuous distributed vector representations for pieces of texts"). Thus, the "paragraph vectors" used by Deliu are a continuous distributed vector representation, as claimed.

165. Indeed, both Deliu and the '900 Patent's inventor, Revanth Patil, in his Master's Thesis (Patil Thesis (EX1020)) cite to Le (EX1019) in describing paragraph vectors. Deliu (EX1004), 50; Patil Thesis (EX1020), 18-19. Le's disclosure ((Le (EX1019), 1, 3) describing paragraph vectors overlaps with Mr. Patil's disclosure (Patil Thesis (EX1020)) in his thesis, which overlaps with the disclosure in the '900 Patent ('900 Patent (EX1001), 5:1-13), which overlaps with Deliu's disclosure ((Deliu (EX1004), 66-67). The following table shows the similarity in disclosures:

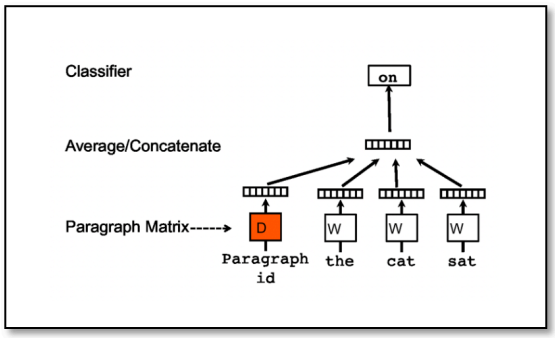
Reference	**Disclosure**
Le (EX1019), 1, 3	"In this paper, we propose Paragraph Vector, an unsupervised framework that **learns continuous distributed vector representations for pieces of texts**. The texts can be of variable-length, ranging from sentences to documents. The name Paragraph Vector is to emphasize the fact that the method can be applied to variable-length pieces of texts, anything from a phrase or sentence to a large document. […] In our Paragraph Vector framework (see Figure 2), **every paragraph is mapped to a unique vector, represented by a column in matrix *D* and every word is also mapped to a**

Reference	Disclosure
	unique vector, represented by a column in matrix *W*. The paragraph vector and word vectors are averaged or concatenated to predict the next word in a context. In the experiments, we use concatenation as the method to combine the vectors. More formally, the only change in this model compared to the word vector framework is in equation 1, where *h* is constructed from *W* and *D*."
Patil Thesis (EX1020)	"Paragraph vector is an supervised framework that **learns continuous distributed vector representations for documents instead of words**. The text in the documents can be of variable-length. As shown in Figure 3.4, **every paragraph is mapped to a unique vector, represented by a column matrix D and every word is also mapped to a unique vector, represented by a column in matrix W. The paragraph vector and word vectors are averaged or concatenated to predict the next word in a context**. The only change in this model compared to word2vec framework is addition of paragraph vector along with the word vectors [Le and Mikolov (2014)]."
'900 Patent (EX1001), 5:1-13	"A paragraph vector is a supervised framework that may be implemented by the system that **learns continuous distributed vector representations for documents instead of words**. The text in the documents can be of variable length. As shown in FIG. 4, **every paragraph is mapped to a unique vector, represented by a column matrix D and every word is also mapped to a unique vector, represented by a column in matrix W. The paragraph vector and word vectors are averaged or concatenated to predict the next word in a context**. The only difference between this Doc2vec PV-DM model compared to the Word2vec framework is the addition of a paragraph vector along with the word vectors."
Deliu (EX1004), 66-67	"The distributed word representation introduced in chapter 2 **maps individual words in fixed length vectors so that similar words are close to each other in the new vector space**. Le and Mikolov in [63] extended that model to be able to represent texts

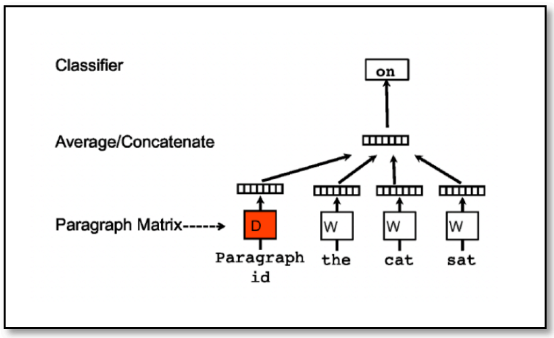
Reference	Disclosure
	of any length. This model, called Paragraph Vectors, follows the same principles as distributed representation for words (word embeddings). That is, **context words are used to predict the next word in the paragraph**. **The only difference is the presence of an additional vector which represents the entire text (paragraph) and is combined with the vectors of context words to predict the next word in the paragraph**. Mathematically, the difference is that in the word embedding model the hidden layer is parametrized only by the context words, so $h = F(W)$; in paragraph vectors, the paragraph matrix D is added as an additional parameter, $h = F(W, D)$."

166. Additionally, each of Le (EX1019), Patil Thesis (EX1020), '900 Patent (EX1001), and Deliu (EX1004) have identical figures illustrating the concept of paragraph vectors, as shown below:

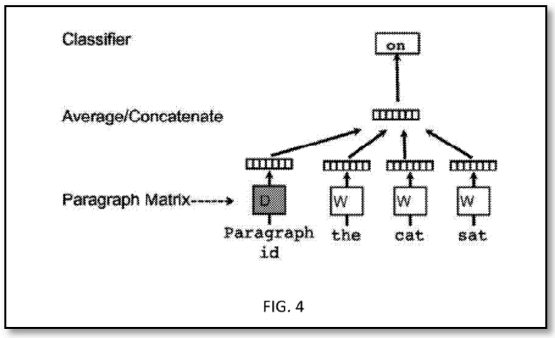
***Le*, Fig. 2**



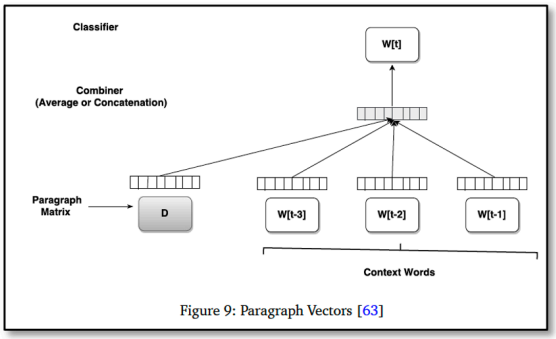
***Patil Thesis*, Fig. 3.4**



'900 Patent, Fig. 4



***Deliu*, Fig. 9**



Le (EX1019), Fig. 2 (top left); Patil Thesis (EX1020), Fig. 3.4 (top right); '900 Patent (EX1001), Fig. 4 (bottom left); Deliu (EX1004), Fig. 9. (All annotations original.)

167. Thus, Deliu teaches the same way as the '900 Patent as to how the "paragraph vectors" are extracted from the data using a neural network and represent continuous distributed vector representations for documents and words within the data.

168. **In sum**, the Deliu-Pendar combination teaches "[t]he computer-implemented system of claim 1, wherein the set of features are extracted from the data using a neural network [Deliu's paragraph vectors, consisting of a neural

network], wherein the neural network is configured to learn a continuous distributed vector representation for documents and words within the data [Deliu's paragraph vectors parameterizing the paragraph and context words]."

F. Claim 12

1. Claim 12(Pre): A method, comprising:

169. To the extent the preamble is limiting, Deliu teaches a method to classify textual data from hacker forums. *See* §VI.A.3 above, Claim 1(Pre); Deliu (EX1004), 7 ("In this thesis, we explore the capabilities of Machine Learning methods in the task of locating relevant threat intelligence from hacker forums").

2. Claim 12(a): configuring a processor for executing operations including:

170. Deliu teaches configuring a processor for executing operations with tools and software. *See* §VI.A.4 above, Claim 1(a).

3. Claim 12(a)(1): accessing data associated with a deep web forum, the data defining a topic for classification;

171. Deliu teaches accessing data from the deep web Nulled.IO forum defining a discussion topic for classification. *See* §VI.A.6 above, Claim 1(a)(2).

172. For example, the Nulled.IO forum is "organized into **topics**, where each topic can have hundred[s] or thousands of posts." *Id.*; Deliu (EX1004), 104. Using "classification methods[,]" Deliu classifies forum posts into categories. Deliu (EX1004), 74-76, Fig. 12.

4. Claim 12(a)(2): extracting a set of features from the data as inputs for a machine classifier; and

173. Deliu teaches extracting features from the Nulled.IO dataset using "Bag-of-Words" and "n-gram[.]" *See* §VI.A.7 above, Claim 1(a)(3).

174. For example, "Bag-of-Words" and "n-gram" extract feature to form word vectors that are used as inputs for Deliu's k-NN, Decision Trees, SVM, and CNN classifiers. Deliu (EX1004), 35 ("Most of the existing machine learning algorithms are able to build predictive models or infer intrinsic structural information **only when the data is presented in the form of feature vectors**."), 41 ("Since machine learning algorithms are able to build predictive models only **from the features extracted** from raw data, their success is closely related to the quality of the features[.]"), 80 (Table 10), 86 (Table 19).

5. Claim 12(a)(3): apply a machine classifier to the set of features to generate a prediction list of tags for classifying the topic, wherein the prediction list includes a prediction probability value for each tag of the plurality of tags; and

175. The Deliu-Pendar combination teaches this limitation. *See* §VI.A.8 above, Claim 1(a)(4).

6. Claim 12(a)(4): adding all parent tags associated with a tag of the plurality of tags to the prediction list based on a comparison between the prediction probability value for the tag and a first predetermined threshold value.

176. The Deliu-Pendar combination teaches this limitation. *See* §VI.A.9 above, Claim 1(a)(5).

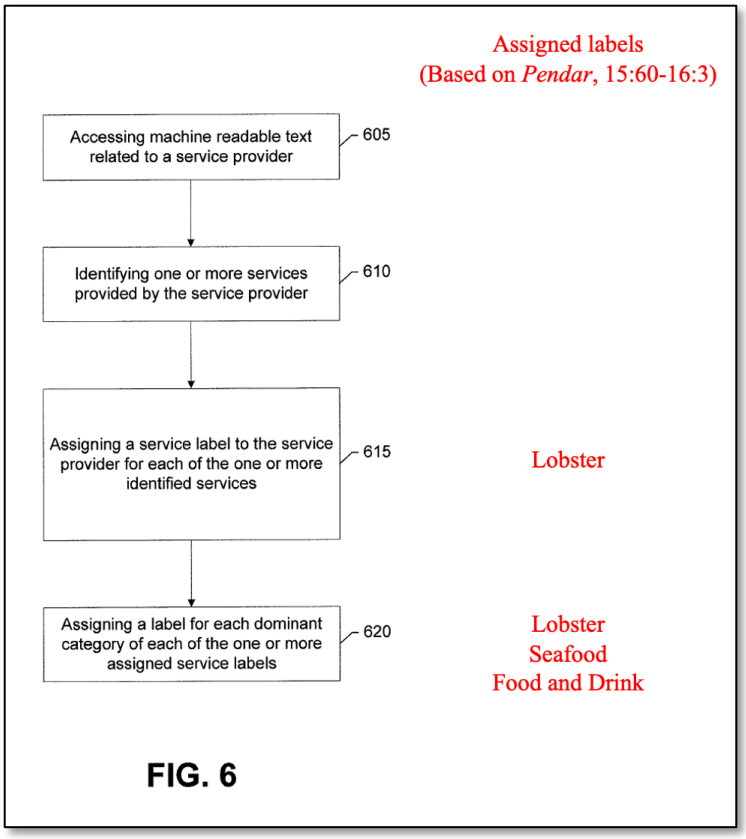
G. Claim 13: The method of claim 12, further comprising: outputting an update to the prediction list of tags, a total number of the tags of the prediction list of tags being different from the update to the prediction list of tags to maintain a predetermined tag hierarchy.

177. The Deliu-Pendar combination teaches the computer-implemented system of claim 12. *See* §VI.F.1 above, Claim 12.

178. The Deliu-Pendar combination further teaches (i) generating a list of labels associated with a document (Pendar, Fig. 6 (step 615) discussed in §VI.A.8 above for Claim 1(a)(4)); and (ii) adding labels for each dominant category of the assigned labels (Pendar, Fig. 6 (step 620) discussed above in §VI.A.9 above for Claim 1(a)(5)).

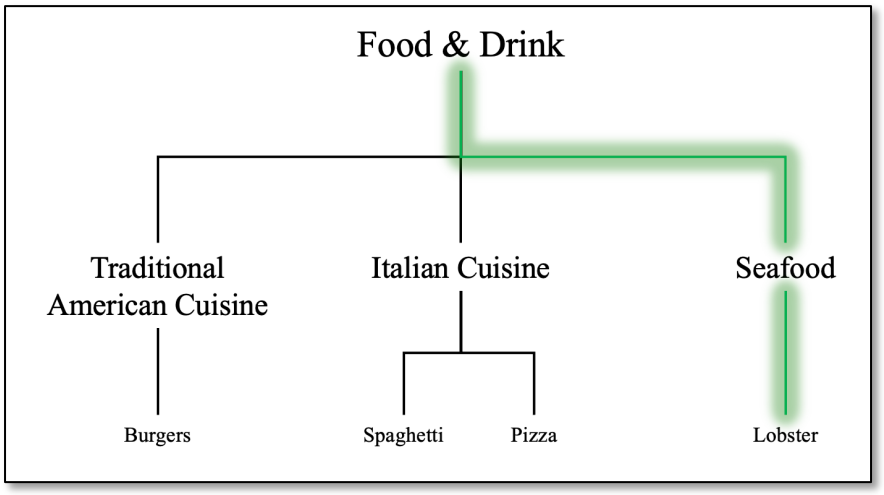
179. In Deliu's modified method, the process of adding dominant labels "updates" the prediction list of labels assigned to a document. Specifically, after Step 615 in Figure 6, a document has, for example, a single label of "lobster[.]" Pendar (EX1005), 15:57-62. After Step 620, two additional labels of "seafood" and "food and drink" have been added. *Id.*, 15:66-16:3 ("As shown in block 620 of FIG. 6, an apparatus, such as a classification and tagging apparatus 100, may be configured for **assigning a label for each dominant category of each of the one or more assigned service labels**. Continuing the example from the previous paragraph, in an instance in which **a label 'lobster' is determined**, the label 'lobster' may be assigned, **a dominant label 'seafood' may be assigned, and a dominant label 'food and drink' may be assigned**."). Thus, there are more assigned labels

(*i.e.*, *a total number of the tags of the prediction list of tags being different from the update to the prediction list of tags*) after the dominant hierarchical labels are added. This process is illustrated in annotated Figure 6 below:



Id., Fig. 6 (annotated), 15:60-16:3.

180. Adding the labels "seafood" and "food and drink" follows Pendar's service taxonomy discussed in §VI.A.5 above for Claim 1(a)(1). (*i.e.*, *to maintain a predetermined tag hierarchy*). This is seen in the annotated path of the figure I created from Deliu's disclosures, shown below:



181. **In sum**, the Deliu-Pendar combination teaches "[t]he method of claim 12, further comprising: outputting an update to the prediction list of tags [Pendar's Step 620 adding all dominant labels], a total number of the tags of the prediction list of tags being different from the update to the prediction list of tags [the number of labels following Pendar's Step 615 increases and changes after Step 620] to maintain a predetermined tag hierarchy [maintains Pendar's service taxonomy]."

VII. OPINION: GROUND 2

A. Obviousness over Deliu, Pendar, and Frank

182. Claim 3 would have been obvious over Deliu, Pendar, and Frank.

1. Overview of Frank

183. I have reviewed the Declaration of Ms. June Munford regarding her methodology in identifying and locating Frank. Munford Dec. (EX1006), ¶¶57-62. As a person with more than ordinary skill in the art, I agree with Ms. Munford's methodology in identifying and locating Frank.

184. In my over 30 years in academia, I have found out that theses are serious, formal, peer-reviewed documents that are often deposited in the institutions' libraries and/or graduate schools. This deposit is crucial for preservation and long-term archival—as well as for verification/confirmation of who defended a thesis successfully, when, and the title/content. I have personally searched, downloaded, and reviewed many theses (including when I was in graduate school myself). *See* §I above.

185. Specifically, I have been asked to opine on whether Frank would have been accessible to person's interested and ordinarily skilled in the subject matter. I have also been asked to consider whether, even without knowledge of the work, the Frank thesis title is adequately descriptive that an individual interested in this research area would have located it through ordinary means. I understand Frank is a Doctoral thesis presented in 2000 at the University of Waikato in Hamilton, New Zealand and was deposited in the University of Waikato library as least as early as January 2001. Munford Dec. (EX1006), ¶¶57-62. I have been made aware of a Declaration from June Munford that describes how interested individuals could have found Frank before the priority date of the '900 Patent. Munford Dec. (EX1006), ¶¶57-62. Ms. Munford describes locating Frank through the University of Waikato's online library catalog and research commons. Namely, Ms. Munford details a number of search queries that produced the Frank thesis in the search results.

Munford Dec. (EX1006), ¶¶61-62. The search queries include a variety of keywords from the title and Abstract of the thesis, and/or author, such as "Pruning Decision Trees and Lists", "Decision Trees", "Decision Trees and Lists", "Pruning Lists", "Eibe Frank," "Pruning Decision Trees," "Pruning Mechanisms", "Machine Learning," "Parametric Tests," and "Practical Data Mining Applications."

186. I agree with Ms. Munford's methodology, and it is therefore my opinion that a person interested in this subject matter could indeed have searched for and discovered the Frank thesis as of 2001. Engineering and technology universities are known to house doctoral theses, including theses in the fields of cyber intelligence and machine learning. An individual interested in these fields, even without prior knowledge of Frank, would have been able to locate the thesis through a reasonable search given the descriptive keywords in Frank's title or other keywords in the body of the thesis.

187. Frank teaches "pruning algorithms for decision trees and lists that are based on significance tests." Frank (EX1007), 3. Frank discusses many pruning methods, including "Critical Value Pruning[.]" *Id*., 76. "Critical value pruning" compares the "splitting criterion" value of a node to a threshold value to make pruning decisions. *Id*. Specifically, "[w]hen a subtree is considered for pruning, the value of the splitting criterion at the corresponding node is compared to a fixed threshold, and the tree is replaced by a leaf if the value is too small." *Id*.

2. Claim 3: The computer-implemented system of claim 1, wherein the processor is configured to remove all child tags of the plurality of tags corresponding to a parent tag from the prediction list of tags where a probability value associated with the child tag is below a second predetermined threshold value.

188. As discussed in §IV.C above, I have adopted Petitioner's interpretation for Claim 3, requesting that the Board correct a typographical error. The corrected claim language is reproduced below:

Claim 3

> "wherein the processor is configured to remove all child tags of the plurality of tags corresponding to a parent tag from the prediction list of tags where a probability value associated with the ~~**child**~~ **[parent]** tag is below a second predetermined threshold value"

189. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1. Specifically, Pendar teaches a summed score (*i.e.*, *a probability value associated with the* ***~~child~~ [parent]*** *tag*). *See* §VI.A.8 above, Claim 1(a)(4).

190. Frank teaches a critical value pruning technique that analyzes hierarchies from the bottom-up. Frank, 76. "When a subtree is considered for pruning, the value of the splitting criterion **at the corresponding node is compared to a fixed threshold**, and **the tree is replaced by a leaf if the value is too small**." *Id.* In other words, a splitting criteria value of a node is compared to a fixed threshold (*i.e.*, *a second predetermined threshold*). If the splitting criterion is below the

threshold (*i.e.*, *below a second predetermined threshold*), then the subtree is removed (*i.e.*, *remove all child tags*) in place of a single leaf node. *Id.*

191. **In sum**, the Pendar-Deliu-Frank combination teaches "[t]he computer-implemented system of claim 1, wherein the processor is configured to remove all child tags [Frank's removing a subtree for a leaf node] of the plurality of tags corresponding to a parent tag from the prediction list of tags where a probability value associated with the [parent] tag [Pendar's summed score] is below a second predetermined threshold value [Frank's fixed threshold for splitting]."

a. Motivation to Combine Deliu-Pendar with Frank

192. A POSITA would have been motivated to combine Deliu and Pendar as discussed in §VI above (Ground 1).

193. A POSITA would have been further motivated to combine the Deliu-Pendar machine classifier including its hierarchical classification techniques with Frank's subtree pruning techniques as taught by Frank for at least the following two reasons.

194. **First**, the Deliu-Pendar combination already teaches summing scores (*i.e.*, *a probability value associated with the [parent] tag*) of each positive query associated with a label. Pendar (EX1005), 14:40-67. Leveraging this summed score to further identify and remove lower-level (*i.e.*, less important) tags that contradict

the natural hierarchy of the dataset would improve classifier performance by reducing false positive predictions of the lower-level tags.

195. **Second**, thresholding is a well-known concept in machine learning, where the probability of making a correct prediction is weighed against a threshold for making an incorrect decision. Smola (EX1021), 30-32 ("Whenever $L(x)$ exceeds a given threshold c we decide to x is spam and consequently reject the e-mail."), Section 1.3. *See also* §V.C above.

196. Hierarchical topologies benefit from probability thresholds, including those associated with dominant levels to ensure compatibility with sub-dominant levels. A POSITA would have looked to ensure compatibility in both a top-down and a bottom-up across all levels of the hierarchy. One known technique to achieve this is to weigh the probability of a predicted dominant label against a threshold to determine whether compliance with the hierarchy is maintained. This would be applying a known technique (*e.g.*, Frank's method of pruning a subtree) to a known device ready for improvement (*e.g.*, Deliu-Pendar's hierarchical classification technique) to yield predictable results (*e.g.*, improved classifier performance). Using the example from Pendar, if the probability of the dominant label "Italian cuisine" is prohibitively low, then pruning the sub-dominant labels ensures that false-positive errors associated with the dominant level are not propagated to lower levels.

197. A POSITA would have had a reasonable expectation of success modifying the Deliu-Pendar combination to use its summed score to compare against Frank's fixed threshold in deciding whether to remove all subtree labels. Frank's critical value pruning is a simple and known technique for improving classification. Because statistics and probability theory "form the language" of machine learning, a POSITA would have expected success in making this modification. Smola (EX1021), 11 ("After that, we will discuss some basic tools from statistics and probability theory, since they form the language in which many machine learning problems must be phrased to become amenable to solving.").

198. Moreover, modifying Deliu-Pendar's datasets with Frank's pruning techniques would have utilized the same hardware and would have required only minor modifications to software. For example, Pendar already sums scores associated with a probability to predict a label, where the summed scores are "checked" against a threshold. Pendar (EX1005), 9:56-62, 14:38-57. Thus, making the minor modifications to software would have been well within the skill of a POSITA at the time.

VIII. OPINION: GROUND 3

A. Obviousness over Deliu, Pendar, Ramírez-Corona

199. Alternatively to Ground 2 (§VII above), to the extent the Board does not correct the typographical error in Claim 3, it would have been obvious over Deliu, Pendar, and Ramírez-Corona.

1. Overview of Ramírez-Corona

200. Ramírez-Corona teaches a hierarchical multi-label classification (HMC) method "for tree and Directed Acyclic Graphs (DAG) hierarchies." Ramírez-Corona (EX1008), 1. Specifically, Ramírez-Corona compares pruning approaches for Non Mandatory Leaf Node Prediction (NMLNP) which varies the pruning direction, time, and condition. *Id*. Ramírez-Corona addresses two different directions for pruning, "top-down" and "bottom-up," as well as different times to perform the pruning stage (*i.e.*, "pruning time") Ramírez-Corona (EX1008), 7. Moreover, Ramírez-Corona uses three pruning conditions when comparing, which "[e]stablish[] the condition to fulfill to prune the node." *Id.*

2. Claim 3: The computer-implemented system of claim 1, wherein the processor is configured to remove all child tags of the plurality of tags corresponding to a parent tag from the prediction list of tags where a probability value associated with the child tag is below a second predetermined threshold value.

201. To the extent the Board does not correct the typographical error in claim 3 and instead reads this claim to require thresholding a "child" tag's probability value against a second threshold, Deliu-Pendar-Ramírez-Corona teaches this claim.

202. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1. Ramírez-Corona teaches a "[h]ierarchical [m]ulti-label [c]lassification" system that uses "pruning" methods when "the information available is not sufficient to estimate the class of an instance at the lower levels in the hierarchy[.]" Ramírez-Corona (EX1008), 1, 6.

203. In these instances, the path of the predicted hierarchy is pruned/truncated. *Id*., 6-7. Ramírez-Corona teaches a bottom-up traversal of the hierarchy, where "[t]he hierarchy is traversed starting from leaf nodes[.]" *Id.*, 7. Ramírez-Corona traverses from the leaf node to its parents, then, "when [a] pruning condition is met in one node, the traversing is stopped and the descendants of the node are pruned." *Id*. This "pruning condition" occurs when "the **probability of the most probable child**" is less than a threshold (*i.e.*, *a probability value associated with the child tag is below a second predetermined threshold value*). *Id*. ("Most probable child (BEST). Prunes if the probability of the most probable child is less than the probability of the 'unknown' label."), 5 ("A multi-class classifier C_i is trained for each non leaf node y_i. The classes in *Ci* are the labels in the set of *child*(y_i) plus an '*unknown*' label that corresponds to the instances that do not belong to any *child*(y_i).").

204. In Ramirez-Corona, if the probability is less than the threshold (*e.g.*, less than the probability of the unknown label), the pruning condition is met, and

"traversing is stopped and the node and **its descendants are pruned**." *Id*., 7. Ramírez-Corona therefore *removes all child tags of the plurality of tags corresponding to a parent tag* when the *probability value associated with the child tag is below a second predetermined threshold value*.

205. **In sum**, the Deliu-Pendar- Ramírez-Corona combination teaches "[t]he computer-implemented system of claim 1, wherein the processor is configured to remove all child tags of the plurality of tags corresponding to a parent tag [Ramírez-Corona's pruning a child] from the prediction list of tags where a probability value associated with the child tag [Ramírez-Corona's probability of most probable child] is below a second predetermined threshold value [Ramírez-Corona's probability of unknown label]."

a. Motivation to Combine Deliu-Pendar with Ramírez-Corona

206. A POSITA would have been motivated to combine Deliu and Pendar as discussed in §VI above (Ground 1).

207. A POSITA would have been motivated to modify Deliu-Pendar's hierarchical multi-level classifiers to implement Ramírez-Corona's pruning technique to maintain coherence in the multi-level tags. Pendar provides a multi-level service taxonomy, for example, an "Italian cuisine" dominant label and "spaghetti" and "pizza" sub-dominant labels. Pendar (EX1005), 15:1-17. Ramírez-

Corona (discloses an algorithm for estimating lower-level classes within a hierarchy. Ramírez-Corona (EX1008), 6.

208. **First,** thresholding is a well-known concept in machine learning, where the probability of making a correct prediction is weighed against a threshold for making an incorrect decision. Smola (EX1021), 30-32 ("Whenever $L(x)$ exceeds a given threshold c we decide to x is spam and consequently reject the e-mail."), Section 1.3. *See also* §V.C above.

209. In the same way, hierarchical topologies benefit from probability thresholds, including for example, those associated with sub-dominant labels to ensure compatibility with the dominant level. A POSITA would have looked to ensure both a top-down and a bottom-up compatibility across all levels of the hierarchy. One known technique to achieve this is to weigh the probability of a predicted sub-dominant label against a threshold to determine whether compliance with the hierarchy is maintained. For example, a POSITA would have understood that a dominant level label of "Italian cuisine" is incompatible with a sub-dominant label of "sushi."

210. **Second,** a POSITA would have looked to the techniques of Ramírez-Corona to ensure that only compatible sub-dominant levels are used. Using the example from Pendar, if the probability of the dominant label "Italian cuisine" is high, but the probability for the sub-dominant label "sushi" does not exceed a

threshold, then pruning the sub-dominant label from consideration ensures compliance with the hierarchy.

211. Indeed, the Deliu-Pendar combination would have appreciated the express benefits discussed in Ramírez-Corona, including providing "superior" classification results with pruning when needed compared to returning complete hierarchical paths for all nodes. Ramírez-Corona (EX1008), 13 ("In every dataset the results obtained by NMLNP version were significantly superior compared to the MLNP version for NMLNP datasets. Thus, pruning obtains better scores than returning complete paths to a leaf node according to the GLB metric.").

212. A POSITA would have had a reasonable expectation of success modifying Deliu-Pendar's multi-level classifiers to implement Ramírez-Corona's pruning technique. Because statistics and probability theory "form the language" of machine learning, a POSITA would have expected success in making this modification. Smola (EX1021), 11 ("…we will discuss some basic tools from statistics and probability theory, since they form the language in which many machine learning problems must be phrased to become amenable to solving").

213. Moreover, modifying Deliu-Pendar's datasets with Ramírez-Corona's pruning technique would have utilized the same hardware and would have required only minor modifications to software. Pendar already sums scores associated with a probability to predict a label, where the summed scores are "checked" against a

threshold. Pendar (EX1005), 9:56-62, 14:38-57. Thus, making the minor modifications to software would have been well within the skill of a POSITA at the time.

IX. OPINION: GROUND 4

A. Obviousness Over Deliu, Pendar, and Chawla

214. Claims 6 and 14 would have been obvious over Deliu, Pendar, and Chawla.

1. Overview of Chawla

215. Chawla teaches an approach to construct "classifiers from imbalanced datasets[.]" Chawla (EX1009), 1. An imbalanced dataset is one in which the "classification categories are not approximately equally represented." *Id.* Chawla's proposed approach is an "over-sampling approach" that oversamples the minority class by generating "synthetic" samples (referred to as "SMOTE"). *Id.*, 6-8.

216. Chawla teaches that the first step to generate synthetic samples is to "[t]ake the difference between the feature vector (sample) under consideration and its nearest neighbor." *Id.,* 8. Then, the difference is multiplied "by a random number between 0 and 1" and added to the feature vector. *Id.* A random point along the line segment between the two features is then selected, and ultimately, this process forces "the decision region of the minority class to become more general." *Id.*

2. *Claim 6: The computer-implemented system of claim 1, wherein a set of synthetic data samples are created for addition to the ground truth dataset to address class imbalance by creating a set of synthetic sample points along a line defined between a first minority feature vector and a second feature vector and adding the set of synthetic sample points to the ground truth dataset.*

217. I understand that Claim 6 as-issued is identical to the originally filed claim 6 in prosecution. '900 File History (EX1002), 30. During prosecution, the Examiner rejected then-pending claim 6 based on Chawla's teachings:

> Chawla et al. is cited to disclose wherein a set of synthetic data samples are created for addition to the ground truth dataset to address class imbalance by creating a set of synthetic sample points along a line defined between a first minority feature vector and a second feature vector and adding the set of synthetic sample points to the ground truth dataset ("We generate synthetic examples in a less application-specific manner by operating "feature space" rather than "data space". The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/ all of the *k* minority class nearest neighbors," Chawla et al., sec. 4.2, 1st para. See also Chawla et al., fig. 3.).

Id., 228-229.

218. Applicant did not amend claim 6 or argue against Chawla's disclosures during prosecution. *Id*., 278-294.

219. As discussed in §§VI.A.3–VI.A.9 above for Claim 1, independent claim 1 should be found unpatentable in view of Deliu-Pendar. Chawla teaches the limitations of Claim 6.

220. Notably, Chawla teaches a near verbatim process for generating synthetic data samples as described in the '900 Patent's specification, for example using the specific "SMOTE" technique. The table below compares those disclosures:

Chawla (EX1009)	**'900 Patent**
"4.2 **SMOTE** We propose an over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with replacement … **The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. Our implementation currently uses five nearest neighbors.** For instance, if the amount of over-sampling needed is 200%, only two neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each. **Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration.**	"Synthetic Minority Over-Sampling Technique (SMOTE) … **In SMOTE, the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen.** For each minority class sample synthetic examples are introduced along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. **The present system currently uses five nearest neighbors. Synthetic samples are generated in the following way:** **1) Take the difference between the feature vector under consideration and its nearest neighbor.** **2) Multiply this difference by a random number between zero and one.**

This causes the selection of a random point along the line segment between two specific features."	**3) Add it to the feature vector under consideration.** In this manner, **a random point is synthetically introduced along the line segment between two specific features**."

Compare, Chawla (EX1009), 8 *with* '900 Patent (EX1001), 7:11-34.

221. **In sum**, the Deliu-Pendar-Chawla combination teaches the "computer-implemented system of claim 1, wherein a set of synthetic data samples are created for addition to the ground truth dataset [Deliu's snowball sampling] to address class imbalance by creating a set of synthetic sample points along a line defined between a first minority feature vector and a second feature vector and adding the set of synthetic sample points to the ground truth dataset [Deliu's snowball sampling modified with Chawla's SMOTE oversampling technique]."

a. Motivation to Combine Deliu-Pendar with Chawla (Synthetic Data Samples)

222. A POSITA would have been motivated and would have found it obvious to implement Chawla's SMOTE technique for dealing with class imbalance in Deliu-Pendar's system to classify data, for at least the following two reasons.

223. As discussed immediately below, Deliu already describes a process for increasing the number of samples in the dataset, but does not address potential "class imbalance" in the increased dataset. Deliu (EX1004), 62. Chawla acknowledges that real-world datasets often do not produce a balance between classes, and teaches

an approach to address class imbalance. Chawla (EX1009), 1 ("Often real-world data sets are predominantly composed of 'normal' examples with only a small percentage of 'abnormal' or 'interesting' examples").

224. **First,** Deliu's machine learning model for forum classification relies on labeled samples in a dataset. Deliu (EX1004), 62. However, the number of samples can be limited, and Deliu describes a process for "[s]nowball sampling" to "increase the number of samples (like a snowball) from the existing samples" for situations where "'recruiting' of samples is difficult[,]" like identifying additional hacker forums. *Id.*, 62. While this increases the overall number of samples, "snowball sampling" does not address class imbalance.

225. Conversely, Chawla states that "[a] dataset is imbalanced if the classification categories are not approximately equally represented[,]" and acknowledges that "real-world data sets are predominantly composed of 'normal' examples with only a small percentage of 'abnormal' or 'interesting' examples. Chawla (EX1009), 1. Chawla thus teaches a method of increasing the number of samples in an imbalanced dataset by synthetically generating samples to add to the dataset. *Id.*, 1.

226. **Second,** a POSITA would have found it obvious to generate samples synthetically to add to a dataset when, as described by Deliu, adding non-synthetic samples is challenging for at least the benefits discussed in Chawla. Chawla

highlights that one benefit of the SMOTE approach is improved "accuracy of classifiers for a minority class." *Id.*, 32. Another benefit of the SMOTE approach is increased coverage of the minority class due to SMOTE providing "more related minority class samples to learn from, thus allowing a learner to carve broader decision regions." *Id.* Thus, synthetically generating samples would have overcome a known imbalance limitation to classifying hacker forum posts.

227. A POSITA would have had a reasonable expectation of success in modifying Deliu-Pendar with Chawla. Deliu already teaches "snowball sampling," "a technique used to increase the number of samples (like a snowball) from the existing samples and it is used in situations where 'recruiting' of samples is difficult." Deliu (EX1004), 62.

228. Implementing SMOTE as taught by Chawla would have only required software modifications to Deliu's current over-sampling techniques, something well within the skill of a POSITA. Indeed, a POSITA would have had success enhancing Deliu's oversampling snowball technique with the functionality in Chawla's oversampling approach, using Chawla's pseudo-code for the SMOTE approach. Chawla (EX1009), 8-9.

3. Claim 14: The method of claim 12, further comprising generating a predetermined tag hierarchy based on a ground truth dataset, and leveraging semi-supervised learning to address class imbalance and increase accuracy of the machine classifier.

229. The Deliu-Pendar combination teaches generating a predetermined tag hierarchy based on a ground truth dataset. *See* §VI.A.5 above, Ground 1, Claim 1(a)(1) (describing hierarchically labeling the ground truth dataset from the Nulled.IO hacker forum). The '900 Patent describes SMOTE as a semi-supervised learning technique. '900 Patent (EX1001), 12:12-13 ("SMOTE semi-supervised learning"); *see also id.*, 11:51-12:13. Chawla describes the same SMOTE technique as described in the '900 Patent and, therefore, "*leverage[es] semi-supervised learning*." *See* §IX.A.2 above, Claim 6.

230. Chawla's SMOTE technique "*address[es] class imbalance and increase[s] accuracy of the machine classifier*." Chawla (EX1009), Abstract ("An approach to the construction of classifiers from **imbalanced datasets** is described … .our method … can achieve **better classifier performance**"), 32 ("The results show that the **SMOTE approach can improve the accuracy** of classifiers for a minority class."), *id.* ("[SMOTE] **improves performance**"), *id.* ("SMOTE provides more related minority class samples to learn from … **leading to more coverage of the minority class**.").

231. **In sum,** the Deliu-Pendar-Chawla combination teaches the "method of claim 12, further comprising generating a predetermined tag hierarchy based on a ground truth dataset [the Deliu-Pendar labels based on Nulled.IO dataset], and leveraging semi-supervised learning to address class imbalance and increase accuracy of the machine classifier [Chawla's SMOTE].

a. Motivation to Combine Deliu-Pendar with Chawla (Tag Hierarchy)

232. For the same reasons discussed in §IX.A.2 above for Claim 6, a POSITA would have been motivated and would have found it obvious to implement Chawla's SMOTE technique for dealing with class imbalance in the Deliu-Pendar classification system. Furthermore, for the same reasons discussed above in Claim 6, a POSITA would have had a reasonable expectation of success making this modification. *Id*.

X. OPINION: GROUND 5

A. Obviousness over Deliu, Pendar, and Xiao

233. Claim 7 would have been obvious over Deliu, Pendar, and Xiao.

1. Overview of Xiao

234. Xiao teaches a system that automatically maps research papers, based on their abstracts, "into patent classification taxonomy, namely International Patent Classification (IPC)." Xiao (EX1010), 1. Xiao's system utilizes a KNN-classifier that calculates "the similarity between an input document and each document in the

training data set." *Id*., 3. In particular, Xiao discusses using five methods to calculate this similarity, and includes the "BM25" weighting method as one of the similarity calculation methods. *Id*.

2. Claim 7: The computer-implemented system of claim 1, wherein the ground truth dataset is supplemented by, for every sample of the ground truth dataset in a minority class, adding top similar documents from the deep web forum information, as identified using an elastic search similarity score, and adding them to the ground truth dataset.

235. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1.

236. Deliu teaches "snowball sampling[,]" "a technique used to increase the number of samples [*i.e.*, *adding [samples] to the ground truth dataset*] (like a snowball) from the existing samples and it is used in situations where recruiting of samples is difficult." Deliu (EX1004), 62 (internal quotations omitted). However, Deliu does not discuss using "an elastic search similarity score" to add "top similar documents" to the ground truth. Xiao teaches this.

237. Notably, Xiao teaches using the "BM25" elastic search technique to identify similar documents—the *same* elastic search technique the applicant described in the specification of the '900 Patent. *Compare*, Xiao (EX1010), 3-4 ("the system calculates the **BM25-based similarity** between the query *q* and each document") *with* '900 Patent (EX1001), 7:54-8:15 ("The elastic search uses BM25 as the default ranking function to provide the relevance score to a given search

query"). Xiao teaches that "BM25 is a widely adopted weighting method in information retrieval[.]" Xiao (EX1010), 3. "In the BM25 weighting scheme, the input document is viewed as a query q containing a serious of terms $\{t_1, t_2, .., t_m\}$. The System collects all the documents in which at least one term occurs. Then the system calculates the BM25-based similarity between the query q and each document d in training data set[.]" *Id.*, 3-4. This calculation is an "elastic search similarity score[.]" Xiao discloses the equation used to calculate the BM25-based similarity between the query q and each document d as the following:

$$Sim_{BM25}(q,d) = \sum_{t \in q} \frac{tf_t^d \cdot (k_1+1) \cdot w_t}{tf_t^d + k_1 \cdot (1-b+b \cdot \frac{dl}{avgdl})}$$

Id., 4.

238. The'900 Patent includes this same equation to calculate "the BM25 score of a document D[,]" as seen below:

$$\text{score}(D, Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f\left(q_i, D + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)\right.}$$

'900 Patent (EX1001), 7:60-65.

239. **In sum,** the Deliu-Pendar-Xiao combination teaches the "computer-implemented system of claim 1, wherein the ground truth dataset [Deliu's Nulled.IO dataset] is supplemented by, for every sample of the ground truth dataset in a minority class, adding top similar documents from the deep web forum information [using Xiao's BM25 technique], as identified using an elastic search similarity score [Xiao's BM25 elastic search outputs similarity score], and adding them to the ground truth dataset [Deliu's similar documents are added to ground truth].

a. Motivation to Combine Deliu-Pendar with Xiao

240. A POSITA would have been motivated to modify Deliu-Pendar to implement Xiao's BM25 elastic search to identify additional similar documents to add to the existing dataset. Dark web forums are known to include a large number of irrelevant posts, a comparatively small number of security-relevant posts, and an even smaller number of posts relevant to specific security categories. Deliu (EX1004), 76, 93-94, Figures 12 (below) and 15. Deliu specifically shows in the figure below that approximately 25% of the posts from the dataset were categorized as "not related to security[,]" and many of the security-related categories individually accounted for 5% of the dataset:

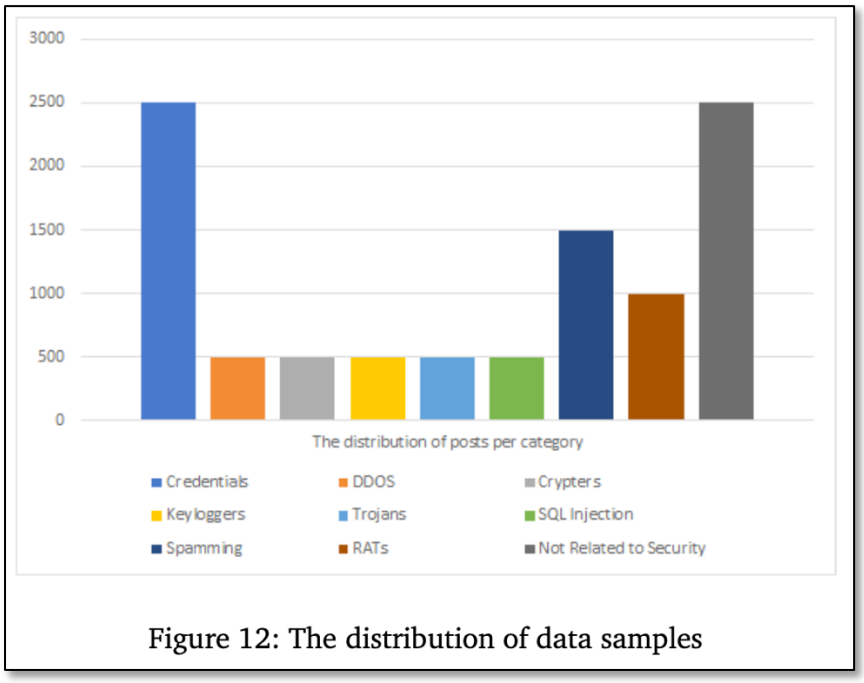


Figure 12: The distribution of data samples

Id., 76, Figure 12.

241. A POSITA would have been motivated to modify Deliu-Pendar with Xiao's BM25 elastic search to increase the size of the less-frequently discussed security categories to improve the classifier's performance in identifying and correctly classifying these documents. Xiao concluded that "BM25 [] outperform[s] other similarity calculation methods." Xiao (EX1010), 6. Thus, applying Xiao's BM25 elastic search to Deliu would increase the accuracy of the classifiers' performance. This would be applying a known technique (*e.g.*, Xiao's BM25 method of identifying similar documents) to a known device ready for improvement (*e.g.*, Deliu's classification system) to yield predictable results (*e.g.*, adding top similar documents to a dataset). Xiao explicitly states, "BM25 is a widely adopted weighting method in information retrieval []." Xiao (EX1010), 3. Therefore, a

POSITA would have been able to apply the known BM25 method of identifying similar documents to Deliu's classification system.

242. A POSITA would have had a reasonable expectation of success in modifying Deliu-Pendar to implement Xiao's BM25 elastic search to identify additional similar documents to add to the existing dataset. Deliu already acknowledges that "recruiting samples" would be difficult and uses a technique called "snowballing" to increase the number of samples. Deliu (EX1004), 62. Xiao teaches the use of an elastic similar score to add top similar documents to a dataset. Xiao (EX1010), 3-4.

243. BM25 is a well-known technique for identifying similar documents. Xiao (EX1010), 3 ("BM25 is a widely adopted weighting method in information retrieval"). BM25 would function in conjunction with Deliu's snowball sampling to increase the number of top ranked similar documents in a dataset, utilizing the same hardware and requiring minor modifications to software that would have been well within the skill of a POSITA at the time. According to Okapi-BM25 Wikipedia Webpage (EX1022), BM25 is used in information retrieval to estimate the relevance of documents to a given search query, and is based on the "probabilistic retrieval framework developed in the 1970s and 1980s []." *Id*., 1. A POSITA would have been aware of this well-known information retrieval technique and would have been able to implement it in Deliu's system.

XI. OPINION: GROUND 6

A. Obviousness over Deliu, Pendar, and Ukrainczyk

244. Claims 8 and 9 would have been obvious over Deliu, Pendar, and Ukrainczyk.

1. Overview of Ukrainczyk

245. Ukrainczyk teaches "a method of simultaneously classifying at least one document into a plurality of categories …." Ukrainczyk (EX1011), 3:27-29. The document being classified is "parsed into a plurality of unique tokens" to create a document feature. *Id.*, Abstract; 3:35-37. A feature vector is created for the document, which includes a count corresponding to each document feature and is ultimately multiplied by a category vector to determine if the document should be classified into a particular category. *Id.*, 3:37-50.

2. Claim 8: The computer-implemented system of claim 1, wherein the data is parsed and segmented into tokens, and predetermined characters and predetermined common words are removed from the data.

246. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1.

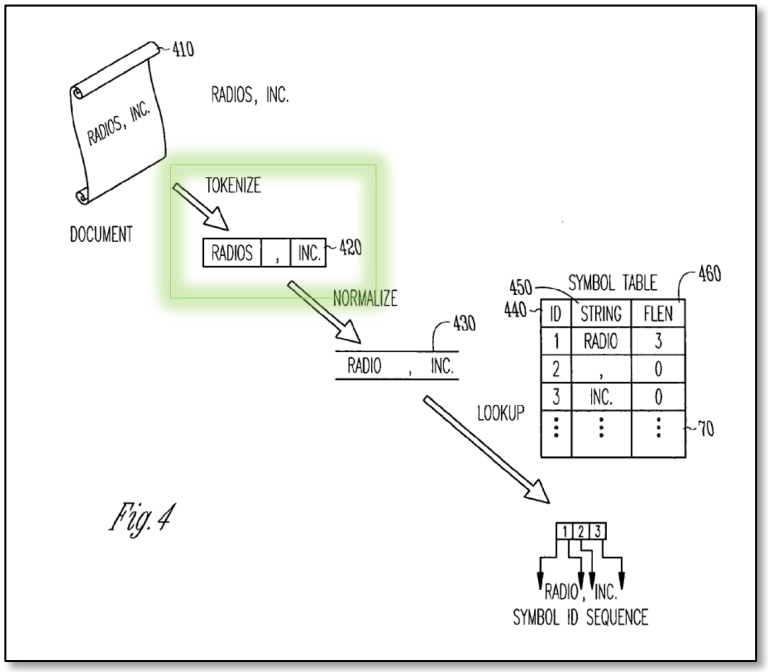
247. Deliu teaches two data preprocessing sub-phases: preparation and cleaning. Deliu (EX1004), 64. Regarding "preparation," "[t]he objective of data preparation is the extraction of only relevant fields from the raw data and their storage in a[n] adequate format … This is especially needed when data is collected

using web crawlers which store the data in a raw HTML format, and special text parsers are used to extract fields such as the content of the posts, title, author, date, etc." *Id*. "On the other hand, the aim of data cleaning is to remove parts of the data which do not contribute to the goal (*e.g.* act as noise), and have a negative effect in the performance of given algorithms." *Id*. (describing "Data cleaning methods" including "**Document Tokenization**").

248. Deliu states that parsing data is "needed[,]" but Deliu's data was already parsed. Deliu (EX1004), 64. As explained in §XI.A.3.a below, a POSITA would have been motivated to implement Deliu's experiments on online hacker forums with data not already stored in relational databases like Nulled.IO. For those online forums, Deliu would need to parse forum data into tokens and would look to known parsing techniques. Ukrainczyk describes one example of known methods for parsing a document into a plurality of unique tokens.

249. Ukrainczyk teaches classifying text into categories, where "tokens or features are … associated with each category." Ukrainczyk (EX1011), Abstract ("a document is parsed into a plurality of unique tokens with associated counts"). In classifying documents, a feature list is constructed including "a list of features automatically extracted from the training set without human intervention." *Id.*, 5:62-63. Features from the feature list are classified into a symbol table after "the text is tokenized[.]" *Id.*, 7:54-55. Ukrainczyk describes a token as "a sequence of

characters that makes up a single unit for processing." *Id.*, 7:55-57. Tokens are full words (*e.g.*, "Yahoo"), punctuation (*e.g.*, "!"), and abbreviations (*e.g.*, "Dr."). *Id.*, 7:57-63. This tokenization process is illustrated in Ukrainczyk's Figure 4 annotated below:



Id., Fig. 4 (annotated).

250. Ukrainczyk further teaches removing certain words and punctuation from consideration. For example, Ukrainczyk identifies stop words, including "the, in, of, at, to, etc." *Id.*, 8:39-41. Stop words are assigned the attribute "Stop[,]" where "if set, the feature is a stop word—it is not counted during topic spotting. This bit is ignored during training[.]" *Id.*, 11:10-11. Ukrainczyk further teaches that "[s]top words, punctuation signs, XML tags, etc. are not passed to feature vector 100[,]" where the feature vector "represents the relevant topic-spotting features within a

segment of text." *Id.*, 12:14-15, 11:43-45. In other words, a POSITA would recognize that Ukrainczyk's methods parse text, tokenize it, then omit certain tokens/words.

251. **In sum**, the Deliu-Pendar-Ukrainczyk combination teaches the "computer-implemented system of claim 1, wherein the data is parsed and segmented into tokens [Ukrainczyk's parsing and tokenization], and predetermined characters and predetermined common words are removed from the data [Ukrainczyk's stop word and punctuation removal].

a. Motivation to Combine Deliu-Pendar with Ukrainczyk

252. A POSITA would have been motivated and would have found it obvious to implement Ukrainczyk's methods for parsing and tokenizing documents and cleaning documents by removing predetermined characters and common words, such as punctuation and stop words, to facilitate Deliu-Pendar's preprocessing sub-phases on online hacker forum datasets, for at least the following reason.

253. Deliu includes "data preparation and cleaning" as its two pre-processing sub-phases. Deliu (EX1004), 64. In the data preparation phase, relevant fields are extracted from the data and stored in an "adequate format[.]" *Id.* In the data cleaning phase, the "noisy" portions of the data are removed to avoid a potential "negative effect in the performance of given algorithms." *Id.* The methods of the

data cleaning subprocess can include processes such as "[l]owercasing[,]" replacing characters, and "[t]okenization. *Id.,* 61, Fig. 8 (annotated below).

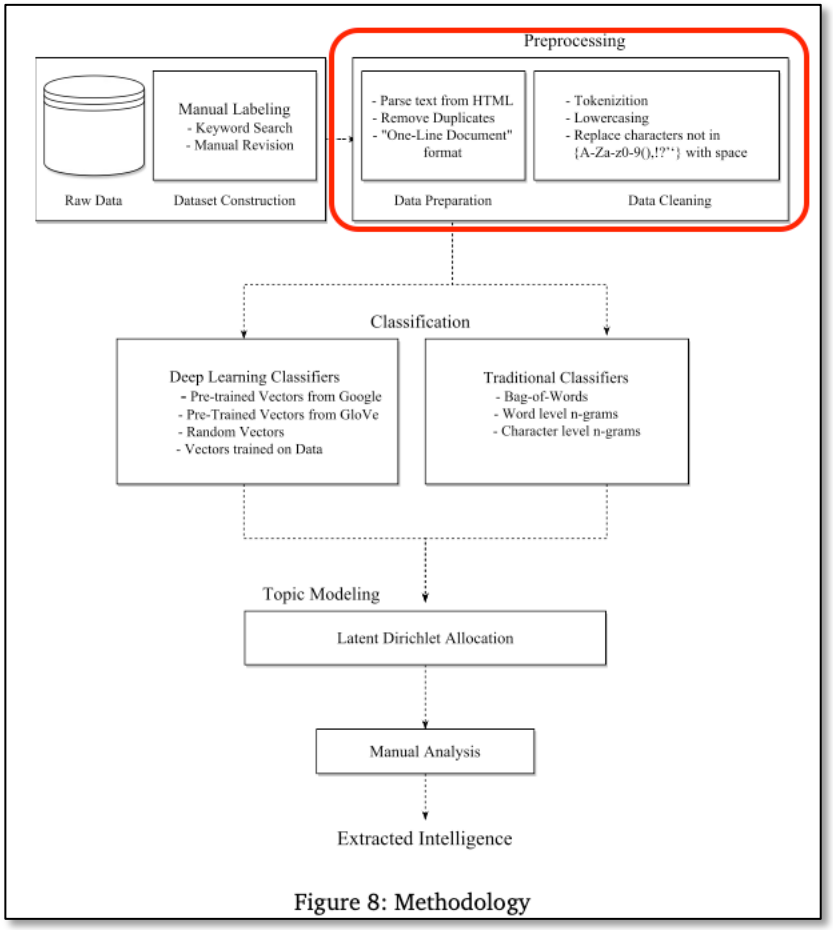


Figure 8: Methodology

Id., Fig. 8 (annotated).

254. A POSITA would have known that tokenization of input text was well known. *See* §V.E.1 above. Tokenization simplifies text by breaking it into manageable tokens that aid in processing and analysis and reduce inconsistencies in language variations. *See e.g.,* Hassler (EX1023), 2 (describing the step of tokenization as segmenting "plain text into appropriate processing units"); *see also id.,* 1 ("To improve the quality of text mining we suggest **linguistically-based**

tokenization as a necessary step preceding further text processing tasks"). Removing predetermined characters and predetermined common words reduces dataset size, speeds up processing, and reduces computation complexity. *See e.g., id.*, 2 ("Thus, **the complexity of subsequent processing tasks (*e.g.*, tagging, chunking) is reduced dramatically** by early made decisions").

255. Removing predetermined characters and predetermined common words also reduces "noise[,]" which enhances classifier accuracy and improves feature extraction. *See, e.g.,* Deliu (EX1004), 64 ("On the other hand, the aim of data cleaning is to remove parts of the data which do not contribute to the goal (*e.g.* act as a noise), and have a negative effect in the performance of given algorithms").

256. A POSITA would have had a reasonable expectation of success modifying Deliu-Pendar to parse and tokenize online hacker forum discussions and clean that tokenized data by removing predetermined characters and common words as taught by Ukrainczyk at least because Deliu already teaches both parsing/tokenizing and data cleaning techniques. Deliu discusses that the techniques for preprocessing data cleaning include "[d]ocument tokenization[,]" "[r]emov[ing] all leading and trailing white-space characters from the document[,]" "[l]owercase text[,]" "[r]eplace characters not in {A-Za-z0-9(),!?'‘} with space[,]" and "[r]eplace two or more consecutive white-space characters with a single white-space[.]" Deliu (EX1004), 64, Table 5. This modification, therefore, would have required simple

programming changes to the techniques already present in Deliu. Deliu already includes that "[d]ocument [t]okenization" is a method of the data cleaning preprocessing phase. *Id.*, 64, Table 5. Implementing Ukrainczyk's tokenization technique in Deliu would yield simplified textual data for processing.

3. Claim 9: The computer-implemented system of claim 1, wherein words within the data are reduced to their root form using stemming and lemmatization procedures.

257. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1.

258. As discussed in §XI.A.2 above for Claim 8, Deliu teaches preprocessing hacker forum data. *See* §XI.A.2 above, Claim 8. While Deliu does not expressly discuss stemming and lemmatization procedures during preprocessing, Ukrainczyk teaches this.

259. Ukrainczyk teaches a stemmer used to "return[] the stem or lemma of the word." Ukrainczyk (EX1011), 8:7-9. When finding word matches, Ukrainczyk analyzes "morphological differences" between two words. *Id.*, 7:35-40, 12:10-12 ("The same two-comparison mechanism is used to check for morphological matches when a feature's Stemming attribute is set."). Indeed, Ukrainczyk considers "plural/singular [differences] and verb inflection[.]" *Id.*, 7:35-40. Ukrainczyk further "relates synonymous terms[.]" *Id.*, 7:43-44.

260. Thus, a POSITA would have understood that Ukrainczyk teaches or suggests stemming and lemmatization. Stemming and lemmatization are considered to be "sibling processes" and are "put under the same roof." Singh (EX1024), 3 ("Stemming **and** lemmatizing are often considered as sibling processes and put under the same roof."). However, the difference between stemming and lemmatization lies in their outputs. *Id.*, 3 ("The basic difference between both processes lies in their outputs."). "The output product of stemming is 'stem' and that of lemmatization is 'lemma'." *Id.*, 3. Thus, a POSITA would have understood that Ukrainczyk's stem and lemma outputs to be the output product of stemming and lemmatization.

261. Indeed, the '900 Patent describes "[l]emmatization" as "the process of grouping together different inflected forms of a word so they can be analyzed as a single item." '900 Patent (EX1001), 9:7-9. Regarding "stemming," the '900 Patent states the "stemming process reduces the derived words—"caresses", "ponies" and "cats" to "caress", "poni", and "cat" respectively." '900 Patent (EX1001), 9:12-14. Ukrainczyk gives similar examples: "carrying" is reduced to "carry," and "dogs" is "dog." Ukrainczyk (EX1011), 8:9-11.

262. Ukrainczyk's lemmatization and stemming techniques reduce words in the dataset to their "normal form[.]" Ukrainczyk, 8:4-6 ("The normalization step takes a token or tag and returns the normal form of the token/tag."). Reducing words

to their normal—or "root"—forms was well known. *See e.g.*, Singh (EX1024), 1 ("Stemming is a program that matches the morphological variants of the word to its root word"); *see also, id.*, 2 ("Stemming is a procedure in which the various morphological variants of words are matched to their stem (root, based word)").

263. **In sum**, the Deliu-Pendar-Ukrainczyk combination teaches the "computer-implemented system of claim 1, wherein words within the data are reduced to their root form using stemming and lemmatization [Ukrainczyk's producing stem and lemma outputs] procedures."

a. Motivation to Combine Deliu-Pendar with Ukrainczyk

264. A POSITA would have been motivated and would have found it obvious to implement Ukrainczyk's stemming and lemmatization data preprocessing techniques in Deliu-Pendar's system to improve classification, for at least the following reason.

265. A POSITA would have known that stemming and lemmatization of input text was well known. *See* §V.E.2 above. Stemming and lemmatization reduce words to their "root forms[,]" which standardizes vocabulary, reduces data dimensionality which improves classifier efficiency, and improves information retrieval by grouping morphologically related words. Singh describes stemming and lemmatizing as "sibling processes" that "perform a similar function of reducing the variant words in the input text." Singh (EX1024), 3. Specifically, Singh highlights

that stemming is one of the known mechanisms to "reduce the size of the index file" and reduce/select features with the goal to "select the most appropriate dimension or class or a concept." *Id.*, 2. These techniques help handle words with complex inflections or irregular forms, including those that would commonly be found in hacker websites. *Id.*, 5 ("These stemmers take into account the morphology of the language while stemming. Inflectional stemmers consider the inflectional morphology *i.e.* they can detect the changes in the word caused due to the syntax such as forms of nouns, verbs, changing singular into plural but the part-of-speech (POS) remains same[]").

266. A POSITA would have had a reasonable expectation of success modifying Deliu-Pendar's preprocessing to further implement stemming and lemmatization because Deliu already teaches preprocessing its datasets, and stemming and lemmatization are widely known and routinely used preprocessing techniques. *See e.g., id.*, 1 ("Stemming is extensively used as a pre-processing tool in the field of natural language processing, information retrieval, and language modeling"); *see also,* §V.E.2 above.

XII. OPINION: GROUND 7

A. Obviousness over Deliu, Pendar, and Nunes

267. Claim 11 would have been obvious over Deliu, Pendar, and Nunes.

1. Overview of Nunes

268. Nunes teaches a "system for cyber threat intelligence gathering from various social platforms on the Internet particularly sites on the dark net[.]" Nunes (EX1012), Abstract. Nunes's system includes a "crawler" that is "a program designed to traverse the website and retrieve HTML documents." *Id.*, 2. The crawlers used by Nunes are specifically designed for different forums. *Id.* ("We have designed separate crawlers for different platforms (markets/forums) identified by experts due to the structural difference and access control measures for each platform").

2. Claim 11: The computer-implemented system of claim 1, further comprising: a crawler in operable communication with the processor and operable to traverse a plurality of deep web websites to retrieve Internet documents forming at least a portion of the deep web forum information.

269. The Deliu-Pendar combination teaches the computer-implemented system of claim 1. *See* §§VI.A.3—VI.A.9 above, Claim 1.

270. Deliu teaches a processor. *See* §VI.A.4 above, Claim 1(a). Deliu further teaches that data collection from hacker forums "demands" web crawlers. While Deliu does not expressly teach *a crawler in operable communication with the processor* because Deliu uses leaked forum data that already exists in a relational database, Deliu suggests web crawlers for use in live data collection from hacker forums. *Id.*, 62. Deliu states, "[t]he data of this forum already exists in a mysql

format and no special web crawlers are needed to collect it." *Id.* Deliu acknowledges that if data from this forum were not publicly available, then special crawlers would be needed to collect the data. *Id.* Deliu does not provide details regarding specific deep/darknet web crawlers for use by its classification system.

271. Nunes teaches a crawler for darknet and deepnet mining of cyber threat intelligence. Nunes (EX1012), Title ("**Darknet and Deepnet** Mining for Proactive Cybersecurity Threat Intelligence"), Abstract ("In this paper, we present an operational system for cyber threat intelligence **gathering from various social platforms on the Internet** particularly sites on the **darknet and deepnet**."). Nunes uses a crawler "to **traverse the website** and **retrieve HTML documents**." *Id.*, 2; *see also id.,* Fig. 1. Nunes crawls a number of websites, including ten darknet/deepnet marketplaces and two darknet forums. *Id.*, 4 ("We consider 10 marketplaces to train and test our learning model."), *id.* ("In addition to the **darknet/deepnet marketplaces** that we have already discussed, there are also numerous **darknet forums** on which users discuss malicious hacking related topics."), *id.* ("We performed evaluation on **two such English forums**.").

272. I note that Nunes is an "arXiv" publicly available technical report that shares two authors who are also inventors listed on the face of the '900 Patent (Paulo Shakarian and Ericsson Marin). Moreover, both references are published by and assigned to the same institution (Arizona State University). Nunes expressly

describes a crawler that operates nearly verbatim to the claim language. Therefore, some inventors of the '900 Patent and their assignee/institution dedicated the concepts claimed in claim 11 to the public back in 2016 when Nunes was published.

273. **In sum,** the Deliu-Pendar-Nunes combination teaches the "computer implemented system of claim 1, further comprising: a crawler in operable communication with the processor and operable to traverse a plurality of deep web websites [Nunes's deepnet crawler] to retrieve Internet documents [Nunes's HTML documents] forming at least a portion of the deep web forum information [Nunes's deepnet crawler retrieving documents from the darknet and deepnet].

a. Motivation to Combine Deliu-Pendar with Nunes

274. A POSITA would have been motivated to implement Nunes's deepnet crawler in Deliu-Pendar to obtain live hacker online forum data for classification. As discussed next, Deliu states that future work should include other experiments that are performed "using other data sources." Deliu (EX1004), 108. Particularly, data sources such as other hacker forums. Nunes teaches a crawler for darknet forums.

275. While Deliu did not use a crawler because the leaked Nulled.IO dataset was already publicly available, Deliu acknowledges that the research "has a commercial development potential." Deliu (EX1004), 107. For example, Deliu envisions extracting cyber-threat intelligence from other hacker forums that have not

been leaked to the Clearnet. *Id.*, 108 ("We are confident that relevant cyber-threat intelligence can also be extracted from hacker forums that use other languages, such as Russian and Chinese … So the scalability of our methods should be tested by extending them to other forums and platforms").

276. A POSITA would have understood that extending the machine learning classification techniques of Deliu to other non-leaked forums would "demand" a crawler to obtain the requisite forum data. *Id.*, 7. Deliu acknowledges that crawlers are needed for the collection of data from hacker forums "due to all the counter-measures enforced by the administrators of such platforms, including: CAPTCHAs, blacklists, registration, invitation only access, etc." *Id.*, 62.

277. A POSITA would have understood that a crawler would have reduced human resources required to identify new forum information, something important for Deliu's intention to scale techniques to a commercial platform. Deliu highlights that for successful commercial application of these techniques, there is a particular importance on the timeliness of the data collection. *Id.*, 106. Specifically, Deliu states, "should these forums be analyzed in real-time then the extracted intelligence will also be timely." *Id.* Therefore, utilizing a crawler would allow the data to be collected in a timely manner rather than utilizing human resources to attempt to access the forums and bypass the forum administrator's counter-measures. *Id.,* 62 ("We must highlight that the collection from hacker forums is challenging; this is

due to all the counter-measures enforced by the administrators of such platforms…"); *see also id.,* 106.

278. A POSITA would have had a reasonable expectation of success implementing Nunes's deepnet crawler to crawl online deepnet hacker forums to retrieve HTML documents. Deliu already acknowledges the need for data collection of "raw data" in constructing a dataset. *Id.*, 61 (Figure 8, "Raw Data"), 62 ("The first phase of analyzing open sources for threat intelligence is the identification and collection of the data."). And while Deliu has access to a leaked dataset, a POSITA would have expected success in obtaining Deliu-Pendar's dataset with a deepnet crawler as taught by Nunes.

279. Deliu establishes the need for specialized crawlers because "data collection demands special web crawlers"; and Nunes describes crawlers that can be employed to be "[t]opic based" and "collect forum discussions from darknet []." Nunes (EX1012), 2. Therefore, Nunes's crawler would have been able to obtain Deliu-Pendar's dataset as it is a specialized, "topic based" crawler that could collect information from the dark web.

280. Web crawlers were prevalent at the time, and special-purpose deepnet crawlers were also well-known. For example, Nunes discusses how topic based crawlers were well known and allowed for focused crawling and data retrieval. *Id.*,

2 (“Topic based crawlers have been used for focused crawling where only webpages of interest are retrieved”).

281. Moreover, Fu describes a crawler that is focused on retrieving data from dark web forums and demonstrates. Fu (EX1025), 1 (“In this study, we propose a novel crawling system designed to collect DarkWeb forum content”); *see* §V.F above. Implementing a known deepnet crawler in Deliu-Pendar would have required no more than the ordinary skill of a skilled artisan. Deliu states that “data collection demands special web crawlers tailored to each source.” Deliu (EX1004), 62. Focused, deepnet crawlers were established and known in the art. Nunes (EX1012), 2 (describing crawlers focused on collecting “forum discussions from darknet); *see* §V.F above. Thus, a POSITA would have been able to implement a crawler tailored for retrieving information from specific dark web forums to access forums that were not yet made public.

XIII. CONCLUSION

282. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: August 25, 2025 By: _____

Erez Zadok, Ph.D