Labeling Hacker Exploits for Proactive Cyber Threat Intelligence: A Deep Transfer Learning Approach

Conference Paper · November 2020

DOI: 10.1109/ISI49825.2020.9280548

5 authors, including:

Sagar Samtani
Indiana University Bloomington
36 PUBLICATIONS **414** CITATIONS
SEE PROFILE

Hongyi Zhu
University of Texas at San Antonio
17 PUBLICATIONS **100** CITATIONS
SEE PROFILE

Steven Ullman
The University of Arizona
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Labeling Hacker Exploits for Proactive Cyber Threat Intelligence: A Deep Transfer Learning Approach

Benjamin Ampel
Department of Management Information Systems
University of Arizona
Tucson, AZ, United States
bampel@email.arizona.edu

Sagar Samtani
Department of Operations and Decision Technologies
Indiana University
Bloomington, IN, United States
ssamtani@iu.edu

Hongyi Zhu
Department of Information Systems and Cyber Security
UTSA
San Antonio, TX, United States
hongyi.zhu@utsa.edu

Steven Ullman
Department of Management Information Systems
University of Arizona
Tucson, AZ, United States
stevenullman@email.arizona.edu

Hsinchun Chen
Department of Management Information Systems
University of Arizona
Tucson, AZ, United States
hsinchun@arizona.edu

***Abstract*—With the rapid development of new technologies, vulnerabilities are at an all-time high. Companies are investing in developing Cyber Threat Intelligence (CTI) to counteract these new vulnerabilities. However, this CTI is generally reactive based on internal data. Hacker forums can provide proactive CTI value through automated analysis of new trends and exploits. One way to identify exploits is by analyzing the source code that is posted on these forums. These source code snippets are often noisy and unlabeled, making standard data labeling techniques ineffective. This study aims to design a novel framework for the automated collection and categorization of hacker forum exploit source code. We propose a deep transfer learning framework, the Deep Transfer Learning for Exploit Labeling (DTL-EL). DTL-EL leverages the learned representation from professional labeled exploits to better generalize to hacker forum exploits. This model classifies the collected hacker forum exploits into eight predefined categories for proactive and timely CTI. The results of this study indicate that DTL-EL outperforms other prominent models in hacker forum literature.**

Index – Hacker forums, cyber threat intelligence, deep transfer learning, text classification, source code, exploit labeling

I. INTRODUCTION

Companies, governments, and academic institutions across the world rely on complex information systems to manage their operations. It is projected that by 2020, there will be 20.8 billion devices connected to the internet [1]. Each of these devices is a potential vector for a cyber-attack, and their rapid development has led to a marked increase in exploitable vulnerabilities. The average cost of a single cybersecurity breach in the United States is $7,010,000 [2]. It is imperative to find ways to proactively identify vulnerabilities and their related exploits before they can be used maliciously. To help protect against such attacks, organizations are investing heavily in developing Cyber Threat Intelligence (CTI). Data is often collected internally from log files, security information and event management systems (SIEMs), and intrusion detection and prevention systems after breaches have already occurred [3], which while valuable, is reactive. Automated data labeling techniques applied to hacker forums can provide proactive CTI, leading to better mitigation techniques for organizations [4]. Exploits can be collected from platforms such as hacker forums and analyzed to help protect against attacks [5]. A sample exploit from one of these platforms is presented in Figure 1.

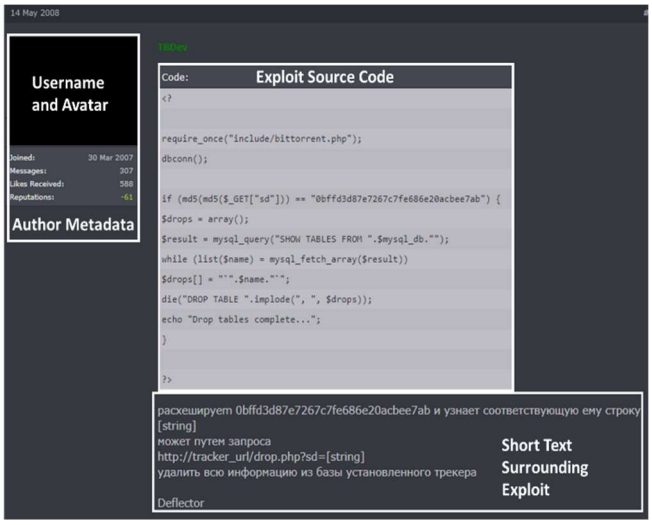

Fig. 1. Example of Exploit Source Code in a Hacker Forum

Traditional hacker forums contain tens of thousands of unlabeled exploit source code that can be used for a potential cyber-attack. However, automated analysis is difficult due to the unreliability of the surrounding text of an exploit. Many exploits can be found as replies to other posts that have little

to do with the thread title, and don't contain meaningful surrounding text.

Exploit specific DarkNet Markets (DNMs) and public exploit repositories contain exploit source code with rich metadata compiled by subject-matter experts, as seen in Figure 2. While these two resources are of great help to the cyber-security community, they are often reactive instead of proactive. Many exploits are only given after the related vulnerability has been patched, making them less useful for proactive CTI efforts.

Fig. 2. Example of Exploit Source Code in an Exploit Specific DNM

In this study, we propose a novel deep transfer learning for exploit labeling (DTL-EL) framework. The DTL-EL framework leverages professionally vetted exploits with a wealth of metadata to improve the performance of hacker forum exploit source code labeling.

The remainder of this paper is organized as follows. First, we discuss prior literature pertaining to hacker forums, text classification, and deep transfer learning. Second, we present our research gaps and questions. Third, we introduce our research design. Finally, we discuss our results, conclusions, and future directions for the work.

II. LITERATURE REVIEW

For this study, three streams of literature are reviewed: (1) hacker forums, (2) text classification, and (3) deep transfer learning. First, we research hacker forums to study the prior methodology and research streams used within the space. Second, we examine text classification techniques in hacker forums to understand how to transform source code and other text features into better predictors for a categorization model. Finally, deep transfer learning is reviewed to discover the most effective way of utilizing public exploit repositories in our DTL-EL framework.

A. Hacker Forums

Hackers use forums, carding shops, DarkNet Marketplaces, and Internet Relay Chat to share goods and assets [6]. On these community platforms, goods are classified as items acquired from a data breach, such as SSNs, usernames, and others, whereas assets are defined as tools used by hackers to facilitate a cyber-attack, like exploits, tutorials, and tools. Hackers congregate at forums to discuss and share assets used to target individuals, organizations, and governments [7], as they are the easiest place to freely share information. Posts on hacker forums can have a significant effect on the occurrence of cyber-attacks [8], meaning there is great societal value in researching them. These forums have millions of text-based posts that contain assets, but posts are noisy and are generally unlabeled. Previous literature on hacker forums has focused on trend identification [9] and exploit categorization [3], [10]–[12]. Of these papers, only Deliu and Williams have used deep learning for their data analysis in Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) models. These papers omit source code when doing analysis, opting to only use post content and author metadata. The other most common methodology applied are support vector machines [5], [10], [11]. As these models are prominent within hacker forum literature, their performance will be used as baselines in our proposed research.

Some research gaps exist in current studies. These papers do not analyze exploit specific DNMs or public repositories, despite including rare metadata like risk levels, CVEs, platforms, lengthy descriptions, and attack labels. These two platforms can potentially be used concurrently with hacker forums due to their similarities and overlapping textual content. To develop a novel approach to label hacker forum exploit code, we require a mechanism to automatically represent text.

B. Text Classification Within Hacker Forums

Bidirectional Long Short-Term Memory (BiLSTM) models are one of the prevailing approaches for categorizing hacker forum text [13]. This is because BiLSTMs models are designed for textual data and can learn embeddings automatically [14]. BiLSTMs models can analyze sequential data from both forward and backward contexts and preserve information from the future and the past. Using the same methodology but inserting a convolutional layer (C-BiLSTM) further improves text classification performance in benchmark tasks [15], as this added layer can extract higher-level phrase representations from the word-embedding layer. Utilizing pre-trained word embeddings, like GloVe, with a C-BiLSTM can improve text classification models even further [16]. GloVe is an unsupervised learning algorithm that obtains the vector representations of words based on their co-occurrence statistics [17]. GloVe word embeddings have been used successfully to improve performance in text, sentiment, and semantic classification tasks [18].

To analyze exploit source code, the best practices for transforming source code into classification model input are

explored. NLP techniques have been used to analyze source code effectively [19], such as utilizing Unicode-based tokenizers as features in predictive models [20]. Utilizing word embeddings for a deep learning source code classification model can also improve performance significantly [21]. For subsequent processing, we could leverage the well-defined metadata contained within exploit specific DNMs and public repositories to aid in our target task of hacker forum exploit source code labeling utilizing transfer learning.

C. Deep Transfer Learning

Deep transfer learning (DTL) aims to improve the performance of a task in a target domain by transferring some type of knowledge from a source domain using a deep neural network architecture [22]. DTL can provide tremendous improvement to classification tasks where the target domain has insufficient data [23]. There are four approaches to DTL: (1) instance-transfer, (2) feature-representation-transfer, (3) parameter-transfer, and (4) relational-knowledge-transfer [24]. Feature-representation is the most suitable for a multi-source, multi-class text classification task, as it finds a good representation of the source features to reduce the difference from the target domain. Both inductive and transductive feature-representation-transfer approaches have been used to significantly outperform state-of-the-art models in benchmark text classification tasks [25]. However, how DTL principles can be leveraged to automatically label hacker forum exploits has not been explored yet.

III. RESEARCH GAPS AND QUESTIONS

Based on our review, we identified the following research gaps. Past exploit categorization in hacker forums is mostly based on post metadata (e.g. title, author, type), but has not directly analyzed source code to create exploit labels. Despite containing professionally vetted exploits with metadata, public exploit repositories have not been leveraged to enhance hacker forum source code labeling. These gaps motivate the following research questions:

- How can we develop a novel deep transfer learning framework that transfers the learned features from public exploit repositories to hacker forums to improve exploit labeling?
- How do deep transfer learning approaches for labeling hacker forum source code compare to non-DTL approaches?

IV. RESEARCH DESIGN

This study aims to create an automated deep transfer learning framework that leverages the rich metadata and labels found in exploit specific DNMs and public repositories to label hacker forum exploit source code. Our research method comprises of four main components (Figure 3): Data Collection, Data Pre-processing, DTL-EL Framework, and Evaluations.

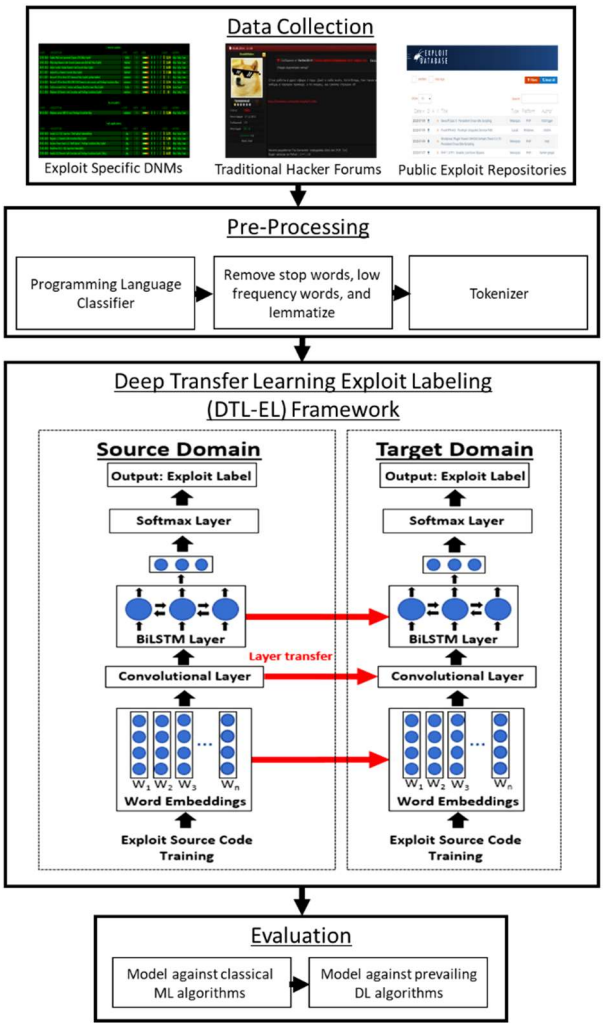

Fig. 3. Proposed Deep Transfer Learning Exploit Labeler Framework

A. Data Collection

Our collection contains three sources of exploits: traditional hacker forums, exploit specific DNMs, and public exploit repositories. Traditional hacker forums and exploit specific DNMs were collected through a crawler routed through the Tor network and parsed immediately upon collection. A depth-first search strategy was implemented for efficient parallel crawling through following different link stacks. This makes the process incremental, as a growing database of previously crawled links and dates is kept for each website, to ensure links are not visited or scraped twice. Public exploit repositories were collected through APIs. All the data was stored in a MySQL database. We summarize our research collection in Table I. Eleven prominent hacker forums were chosen for several reasons. First, they are well

known in the hacker community to discuss hacker assets. Second, they did not require special registration. Finally, these forums encompass a large global user base.

TABLE I. DATA COLLECTION

Platform Type	Name	Language	Posts	Source Code
Traditional Hacker Forums	0x00sec	English	9,161	397
	Altenens	English	1,261,435	1,403
	AntiChat	Russian	2,492,497	64,890
	AntiOnline	English	291,914	2,063
	Ciphers	English	51,612	2,207
	Exelab	Russian	105,312	3,597
	ExeTools	English	45,834	1,832
	go4expert	English	62,103	5,800
	KernelMode	English	29,755	934
	WWHClub	Russian	1,492,156	53
	WildersSecurity	English	2,571,053	2,096
Exploit Specific DNMs	0day.today	English	33,766	33,766
Public Repositories	Seebug	English	56,657	56,657
	ExploitDB	English	43,120	43,120
	PacketStorm	English	39,433	39,433
	Metasploit	English	4,040	4,040
	Vulnerlab	English	1,635	1,635
	Zeroscience	English	651	651
Total:	**18 Sources**	**EN / RU**	**8,592,134**	**264,574**

Source code was not collected in the database unless it was longer than 100 characters to avoid collecting uninformative material. In total, 85,272 exploit source code snippets were collected from hacker forums across 8,412,832 posts, 33,766 from exploit specific DNMs, and 145,536 from public repositories. All source code from exploit specific DNMs and public repositories had labeled attack type, while none were labeled in hacker forums.

B. *Data Pre-Processing*

Upon collection, all source code is run through a machine learning classifier to identify the programming language of the exploit. We then kept the eight most popular exploit categories based on attack type: web applications, denial of service (DoS), remote, local, SQL injection, cross-site scripting (XSS), file inclusion, and overflow. Source code was stripped of unnecessary symbols, made lower-case, lemmatized, tokenized, and then put through a sequence padder to ensure proper lengths for all inputs. For exploit specific DNMs and public exploit repositories, the title metadata is concatenated to the end of the source code.

C. *DTL-EL Framework*

As seen in Figure 3, our framework makes use of two C-BiLSTM models. The model in the source domain is trained on the exploit source code collected from exploit specific DNMs and public repositories. Layers were chosen based on best text classification performances within the literature [26]. First, the input is embedded using the weights of an embedding matrix provided by GloVe. Second, the embedding is passed through a convolutional layer with a kernel size of 3 using a rectified linear unit (ReLU) for activation. Third, it is passed through a one-dimensional max-pooling layer, and a subsequent dropout layer, and then into a BiLSTM layer. Finally, a dense layer of the size of our eight chosen outputs uses a softmax activation function, appropriate for multi-class text classification tasks. The model is trained and saved for future use.

Our DTL-EL is a C-BiLSTM created in the target domain. The DTL-EL uses the transferred embedding, convolutional, and BiLSTM layers from the source domain model before training on exploit source code collected from hacker forums. The transferred layers were chosen based on ablation analysis to find the best performance among layers.

D. *Evaluation*

DTL-EL is evaluated with two sets of experiments. The first experiment compares it against leading classification methods on the source domain. The second compares DTL-EL against non-DTL approaches on the target domain. Experiments were chosen based on prevailing practices in DTL text classification literature [27][28]. Both experiments use accuracy, F1, precision, and recall as metrics. 10-fold cross-validation is used for each model with the same split to allow for significance comparisons across folds. Paired t-tests are used to evaluate statistically significant differences between our proposed approach and benchmarks.

Executing each experiment requires a gold-standard dataset. For DTL research, a source and target dataset are required. To this end, we present a summary of the row count for each exploit label in each domain in Table II.

TABLE II. SOURCE AND DOMAIN DATA TESTBEDS

Exploit Label	Source Domain Count	Target Domain Count
Web Applications	43,475	57
DoS	12,121	714
Remote	11,787	672
Local	7,993	1,952
SQL injection	7,187	702
XSS	7,025	485
File inclusion	3,412	29
Overflow	3,333	231
Total	**96,333**	**4,842**

In total, 101,175 exploits are used across both datasets. The source domain dataset contains 96,333 exploits and was labeled by professional subject-matter experts in their respective source. The target domain contains 4,842 exploits and was manually labeled by our team using keyword searching and row-by-row checking. Web application exploits account for 45.13% of the exploits in the source

domain, making it the most prominent category and our baseline. Local exploits are 40.32% of the target domain.

V. RESULTS AND DISCUSSION

To create the best overall model for identifying hacker exploits, we set up two experiments. The first focuses on creating the best model for labeling professionally vetted exploits from repositories and exploit DNMs, known as the source domain. The second focuses on transferring layers from that model to the target domain to label hacker forum exploit source code. Finally, we show a specific example of our DTL-EL model against the best non-DTL-based model.

A. Experiment 1: Source Domain Training

Our DTL-EL model is tested against prevailing deep learning (e.g. LSTM, GRU, RNN) and classical machine learning algorithms (e.g. SVM, gradient boosted decision tree, logistic regression, Naïve Bayes) in literature. Model performance is evaluated on accuracy, precision, recall, and F1-score. Table 3 summarizes the results of the source domain model training. The top-performing algorithm for each metric appears in boldface.

TABLE III. EXPERIMENT 1: SOURCE DOMAIN RESULTS

Model	Results			
	Accuracy	Recall	Precision	F1
C-BiLSTM (DTL-EL)	**87.38%**	87.18%	**87.64%**	**87.41%**
LSTM	87.32%	**87.26%**	86.80% *	87.03% *
GRU	87.04% **	86.88% **	87.28% **	87.07% **
RNN	80.12% ***	78.31% ***	82.01% ***	80.07% ***
SVM	78.21% ***	77.02% ***	79.86% ***	78.43% ***
XGBoost	70.55% ***	67.51% ***	70.83% ***	69.17% ***
Log Reg	66.11% ***	71.02% ***	66.68% ***	65.84% ***
Naïve Bayes	58.06% ***	56.79% ***	57.93% ***	57.36% ***

* indicates a statistically significant difference at $p < 0.05$, ** at $p < 0.01$, and *** at $p < 0.001$. Based on the results of experiment 1, the C-BiLSTM achieves the highest performance in accuracy at 87.37%, F1-score at 87.41%, and precision at 87.64%. In recall, it performs worse than an LSTM by only 0.08%, but our model performs significantly better than all other tested benchmark models in precision and the comprehensive F1-score metric. Given its superior performance, we choose to use the C-BiLSTM model to transfer layers to our DTL-EL model in the target task.

B. Experiment 2: Target Domain Training

Like the prior experiment, our DTL-EL model is tested against prevailing models in literature that do not use transfer learning, and evaluated on the same four metrics. We summarize the evaluation results in Table 4. The top-performing model performances are highlighted in boldface. Asterisks indicate statistically significant differences.

TABLE IV. EXPERIMENT 2: TARGET DOMAIN RESULTS

Model	Results			
	Accuracy	Recall	Precision	F1
DTL-EL	**65.88%**	**64.35%**	**69.32%**	**66.07%**
C-BiLSTM	63.05% ***	59.71% ***	67.56% ***	63.21% ***
LSTM	62.39% ***	60.49% ***	65.77% ***	63.42% ***
GRU	61.34% ***	59.27% ***	64.06% ***	62.09% ***
RNN	57.64% ***	53.93% ***	62.89% ***	57.62% ***
SVM	48.72% ***	27.38% ***	37.98% ***	32.68% ***
XGBoost	47.65% ***	30.06% ***	48.87% ***	38.97% ***
Log Reg	37.16% ***	38.85% ***	35.13% ***	36.99% ***
Naïve Bayes	8.59% ***	15.08% ***	18.09% ***	16.58% ***

The results from experiment 2 suggest that DTL-EL's transferred layers can better generalize to hacker forum source code than each prominent deep learning and classical learning approach. DTL-EL leads to statistically significant performance increases in accuracy at a 3.22% increase, F1-score at a 3.43% increase, precision at a 1.47% increase, and recall at a 4.56% increase over the C-BiLSTM without any transferred layers. The non-DTL-based models also have a much greater difference between training and validation metric scores than DTL-EL, suggesting they are overfitting. Comparing examples from these models allows us to see specific differences in how they label exploits. Figure 4 shows an exploit snippet from our hacker forum data, along with the correct label, the DTL-EL label, and the non-DTL C-BiLSTM label.

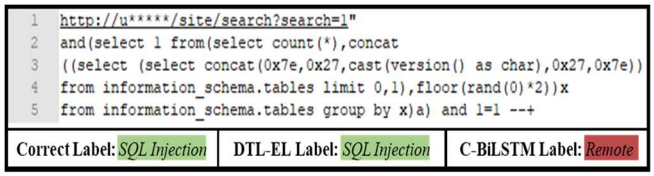

Fig. 4. Example Exploit Labeling

This SQL injection example was chosen due to being easily identified by subject-matter experts as the correct label. The DTL-EL model labeled this correctly with a softmax probability of 0.98, while the C-BiLSTM model considered it to be a remote exploit with a 0.92 probability. This suggests that our model learns features in the exploit source code that the non-DTL-based C-BiLSTM does not.

VI. CONCLUSION & FUTURE WORK

In this study, we aimed to develop a novel approach that provides valuable information about hackers and emerging exploits through a deep transfer learning framework. Our results indicate that the DTL-EL framework offers a significant benefit to labeling hacker exploit source code over baseline non-DTL techniques. Our research can lead to proactive CTI and provide invaluable information to organizations about how to focus their cybersecurity efforts. The DTL-EL model can be applied to collected hacker source code immediately upon collection to build trend analysis

charts and prominent hacker networks, which could create benefits for organizations in protecting their infrastructure.

Future work can focus on how to better utilize collected metadata in the source domain to improve performance, ablation studies for hyperparameter tuning and DTL layer transfer, and implementation of an attention mechanism for increased metric performance and explainable results. This model also has the potential for being used on different types of hacker datasets to identify trends in personal identifiable information leaks, tutorials, and other cybersecurity topics.

VII. ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grant numbers DUE-1303362 (SFS), OAC-1917117 (CICI), and CNS-1850362 (SaTC CRII).

REFERENCES

[1] I. Jang, D. Lee, J. Choi, and Y. S. Son, "Knowledge of Things: A novel approach to share self-taught knowledge between IoT devices," *2018 IEEE International Conference on Consumer Electronics, ICCE 2018*, vol. 2018, pp. 1–2, 2018

[2] J. E. Lerums, L. D. Poe, and J. E. Dietz, "Simulation Modeling Cyber Threats, Risks, and Prevention Costs," *IEEE International Conference on Electro Information Technology*, vol. 2018-May, pp. 96–101, 2018

[3] R. Williams, S. Samtani, M. Patton, and H. Chen, "Incremental hacker forum exploit collection and classification for proactive cyber threat intelligence: An exploratory study," *2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018*, pp. 94–99, 2018

[4] J. Grisham, S. Samtani, M. Patton, and H. Chen, "Identifying mobile malware and key threat actors in online hacker forums for proactive cyber threat intelligence," *2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data, ISI 2017*, pp. 13–18, 2017

[5] S. Samtani, R. Chinn, H. Chen, and J. F. Nunamaker, "Exploring Emerging Hacker Assets and Key Hackers for Proactive Cyber Threat Intelligence," *Journal of Management Information Systems*, vol. 34, no. 4, pp. 1023–1053, 2017

[6] V. Benjamin, W. Li, T. Holt, and H. Chen, "Exploring threats and vulnerabilities in hacker web: Forums, IRC and carding shops," *2015 IEEE International Conference on Intelligence and Security Informatics: Securing the World through an Alignment of Technology, Intelligence, Humans and Organizations, ISI 2015*, pp. 85–90, 2015

[7] V. Benjamin, J. S. Valacich, and H. Chen, "DICE-E: A framework for conducting Darknet identification, collection, evaluation with ethics," *MIS Quarterly: Management Information Systems*, vol. 43, no. 1, pp. 1–22, 2019

[8] W. T. Yue, Q. H. Wang, and K. L. Hui, "See no evil, hear no evil? Dissecting the impact of online hacker forums," *MIS Quarterly: Management Information Systems*, vol. 43, no. 1, pp. 73–95, 2019

[9] M. Schafer, M. Fuchs, M. Strohmeier, M. Engel, M. Liechti, and V. Lenders, "BlackWidow: Monitoring the Dark Web for Cyber Security Information," *International Conference on Cyber Conflict, CYCON*, vol. 2019-May, pp. 1–21, 2019

[10] I. Deliu, C. Leichter, and K. Franke, "Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks," *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, vol. 2018-Janua, no. iii, pp. 3648–3656, 2018

[11] I. Deliu, C. Leichter, and K. Franke, "Collecting Cyber Threat Intelligence from Hacker Forums via a Two-Stage, Hybrid Process using Support Vector Machines and Latent Dirichlet Allocation," *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, pp. 5008–5013, 2019

[12] S. Samtani, R. Chinn, and H. Chen, "Exploring hacker assets in underground forums," *2015 IEEE International Conference on Intelligence and Security Informatics: Securing the World through an Alignment of Technology, Intelligence, Humans and Organizations, ISI 2015*, pp. 31–36, May 2015

[13] M. Ebrahimi, M. Surdeanu, and H. Chen, "Detecting cyber threats in non-english dark net markets: A cross-lingual transfer learning approach," 2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018, pp. 85–90, 2018

[14] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling," *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*, vol. 2, no. 1, pp. 3485–3495, 2016.

[15] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019

[16] R. A. Stein, P. A. Jaques, and J. F. Valiati, "An analysis of hierarchical text classification using word embeddings," *Information Sciences*, vol. 471, pp. 216–232, 2019

[17] M. Pennington, Jeffrey; Richard Socher; Christopher, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[18] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4: Sentiment Analysis in Twitter," *In Proceedings of the 11th international workshop on semantic evaluation,* pp. 502–518, 2018

[19] A. Hindle, E. T. Barr, M. Gabel, Z. Su, and P. Devanbu, "On the naturalness of software," *Communications of the ACM*, vol. 59, no. 5, pp. 122–131, 2016

[20] M. Jimenez, C. Maxime, Y. le Traon, and M. Papadakis, "On the impact of tokenizer and parameters on n-gram based code analysis," *Proceedings - 2018 IEEE International Conference on Software Maintenance and Evolution, ICSME 2018*, pp. 437–448, 2018

[21] A. Leclair, Z. Eberhart, and C. McMillan, "Adapting neural text classification for improved software categorization," *Proceedings - 2018 IEEE International Conference on Software Maintenance and Evolution, ICSME 2018*, pp. 461–472, 2018

[22] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2015-Janua, no. Ijcai, pp. 4119–4125, 2015.

[23] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, *A survey of transfer learning*, vol. 3, no. 1. Springer International Publishing, 2016.

[24] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11141 LNCS, pp. 270–279, 2018

[25] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, pp. 328–339, 2018

[26] A. Yenter and A. Verma, "Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis," *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2017*, vol. 2018-Janua, pp. 540–546, 2017

[27] T. Semwal, G. Mathur, P. Yenigalla, and S. B. Nair, "A practitioners' guide to transfer learning for text classification using convolutional neural networks," *SIAM International Conference on Data Mining, SDM 2018*, pp. 513–521, 2018

[28] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning Based Text Classification: A Comprehensive Review," *arXiv preprint arXiv:2004.03705*., 2020