BUSTECH 2020

The Tenth International Conference on Business Intelligence and Technology

ISBN: 978-1-61208-785-6

October 25 - 29, 2020

BUSTECH 2020 Editors

Jaime Lloret Mauri, Universitat Politecnica de Valencia, Spain

Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster (WWU) /
DIMF / Leibniz Universität Hannover, Germany

BUSTECH 2020

Forward

The Tenth International Conference on Business Intelligence and Technology (BUSTECH 2020), held on October 25 - 29, 2020, continued a series of events covering topics related to business process management and intelligence, integration and interoperability of different approaches, technology-oriented business solutions and specific features to be considered in business/technology development.

The conference had the following tracks:

- Modeling and simulation
- BPM and Intelligence
- Information Technology-enabled Organizational Transformation

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the BUSTECH 2020 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to BUSTECH 2020. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the BUSTECH 2020 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope BUSTECH 2020 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of business intelligence and technology.

BUSTECH 2020 Steering Committee

Malgorzata Pankowska, University of Economics in Katowice, Poland
Jürgen Sauer, Universität Oldenburg, Germany
Pierre Hadaya, ESG-UQAM, Canada
Hans-Friedrich Witschel, University of Applied Sciences and Arts, Northwestern Switzerland
Anbang Xu, IBM Research – Almaden, USA

BUSTECH 2020 Publicity Chair
Javier Rocher, Universitat Politecnica de Valencia, Spain

BUSTECH 2020

Committee

BUSTECH 2020 Steering Committee

Malgorzata Pankowska, University of Economics in Katowice, Poland
Jürgen Sauer, Universität Oldenburg, Germany
Pierre Hadaya, ESG-UQAM, Canada
Hans-Friedrich Witschel, University of Applied Sciences and Arts, Northwestern Switzerland
Anbang Xu, IBM Research – Almaden, USA

BUSTECH 2020 Publicity Chair

Javier Rocher, Universitat Politecnica de Valencia, Spain

BUSTECH 2020 Technical Program Committee

Gulsum Akkuzu, University of Portsmouth, UK
Unal Aksu, Utrecht University, Netherlands
Reza Barkhi, Virginia Tech, USA
Khouloud Boukadi, Faculty of Economics and Management of Sfax, Tunisia
Karl Cox, University of Brighton, UK
Beata Czarnacka-Chrobot, Warsaw School of Economics, Poland
Giuseppe A. Di Lucca, University of Sannio - RCOST (Research Center on Software Technology), Italy
António Dourado, University of Coimbra, Portugal
Helena Dudycz, Wroclaw University of Economics, Poland
Johannes Edler, University of Applied Sciences Upper Austria Campus Hagenberg, Austria
Sean Eom, Southeast Missouri State University, USA
Bedilia Estrada-Torres, Universidad de Sevilla, Spain
Jiri Feuerlicht, University of Economics, Prague, Czech Republic
Lixin Fu, University of North Carolina at Greensboro, USA
Todorka Glushkova, Paisii Hilendarski University of Plovdiv, Bulgaria
Manuel Gomez-Olmedo, Universidad de Granada, Spain
Foteini Grivokostopoulou, University of Patras, Greece
Pierre Hadaya, ESG-UQAM, Canada
Rawad Hammad, University of East London, UK
Mariem Haoues, Mir@cl Laboratory - University of Sfax, Tunisia
Uswatun Hasanah, Universitas Amikom Purwokerto, Indonesia
Ioannis Hatzilygeroudis, University of Patras, Greece
Wladyslaw Homenda, Warsaw University of Technology, Poland
Wei-Chiang Hong, School of Computer Science and Technology - Jiangsu Normal University, China
Wassim Jaziri, Taibah University, Saudi Arabia
Maria João Ferreira, Universidade Portucalense, Portugal
Jānis Kampars, Riga Technical University, Latvia

Copyright Information

Table of Contents

Challenge AI's Mind: A Crowd System for Proactive AI Testing

Siwei Fu

Zhejiang Lab
1818 Wenyi West Road, Hangzhou, Zhejiang, China
Email: fusiwei339@gmail.com

Xiaotong Liu, Anbang Xu, Rama Akkiraju

IBM Research-Almaden
650 Harry Rd, San Jose, California 95120, United States
Email: xiaotong.liu@ibm.com

***Abstract*—Artificial Intelligence (AI) has burrowed into our lives in various aspects; however, without appropriate testing, deployed AI systems are often being criticized to fail in critical and embarrassing cases. In this paper, we propose the concept of proactive testing to dynamically generate testing data and evaluate the performance of AI systems. We further introduce Challenge.AI, a new crowd system that features the integration of crowdsourcing and machine learning techniques in the process of error generation, error validation, error categorization, and error analysis. The evaluation shows that the crowd workflow is more effective with the help of machine learning techniques.**

Keywords–Crowdsourcing; Artificial Intelligence; Proactive Testing.

I. Introduction

Artificial Intelligence (AI) becomes a technology renaissance and is beginning to solve problems in many domains. It often performs well under single-score metrics such as precision and recall. Yet, with all of the AI success, many AI applications are also criticized as they can fail in critical and embarrassing cases. For example, recent AI-powered facial recognition systems of Microsoft, IBM, and Face++ have 34% more errors with dark-skinned females than light-skinned males [1].

To address this problem, we propose *proactive testing*, a novel approach that evaluates the performance of AI models with dynamic and well-crafted dataset collected using crowd intelligence. Proactive testing differs from conventional testing metrics in two aspects. First, it extends the coverage of the testing dataset by dynamically collecting external dataset. Second, AI developers are allowed to query additional dataset belonging to certain categories to target corner cases. As a result, proactive testing is an approach to discovering unknown error and bias of a model, and providing a comprehensive evaluation of the model's performance regarding all test cases.

In this paper, we contribute a hybrid system, Challenge.AI, that combines human intelligence and machine learning techniques to assist AI developers in the process of proactive testing. Our system contains four main components including explanation-based error generation, error validation, categorization, and analysis. We bring in crowd force in error generation and encourage the crowd to craft sentences that can fail a given AI model. Especially, to assist error generation, we borrow advanced machine learning methods to explain each prediction made by the model, and present the explanation to the crowd using intuitive visualization. In addition, we employ the crowd in error validation and categorization to ensure the quality of the crafted dataset at scale. We evaluate the effectiveness of explanation-based error generation by measuring the performance of the crowd. The evaluation shows that the crowd spent less time in generating specific errors.

The structure of the rest of this paper is as follows: Section 2 discusses the related works and techniques. In Section 3 we describe a formative study to outline the challenges of model testing. Section 4 presents our Challenge.AI system, followed by an evaluation of error generation with the crowd in Section 5. We report the evaluation results of our Challenge.AI system with AI developers in Section 6, and discuss design implications in Section 7. We conclude the paper in Section 8.

II. Related Work

This paper is related to prior work in three areas, e.g., generation of adversarial samples using machine learning, acquisition of corpus using crowd intelligence, and the effects of various prompts.

A. Adversarial learning for text classifiers

Several approaches have been proposed to generate adversarial examples in the deep learning community. However, most studies have focused on attacking image or audio classification models [2], [3], [4], [5], [6]. The attack of text classifiers is under-exploited due to the discrete domains involved in text [7].

To craft adversarial samples for text classifiers, some works modify the original input. For example, Liang et al. [8] proposed three perturbation strategies, e.g., insertion, deletion, and replacement to evade DNN-based text classifiers. Li et al. [9] studied the effect of removal of input text at different levels of representation. Gao et al. [10] proposed novel scoring strategies to identify critical tokens and executed a modification on those tokens. Similarly, HotFlip et al. [11] edited the input text at the character level. Ribeiro et al.[12] furthered the research by manipulating the input at the word level. That is, replacing tokens by random words of the same POS (part-of-speech) tag. Given access to the model's architecture, e.g., the computational graph, Papernot et al. [13] manipulated the output of RNN models. Although aforementioned approaches can generate sentences that fail text classifiers, the perturbation harms text integrity, resulting in unnatural and semantically meaningless text from language viewpoint.

To overcome the limitation of above methods, Samanta et al. [14] proposed a rule-based approach to ensure that the resulting text is syntactically correct. Zhao et al. [7] proposed GAN-based approach to generate adversarial input that are legible to humans. The two techniques driven by machine learning are promising in their scalability. However, the resulting text has not been validated, and its quality is not guaranteed. In this paper, we design a crowdsourcing pipeline to generate and validate adversarial samples by means of human intelligence. The derived adversarial dataset is diverse from different perspectives.

B. Corpus acquisition using crowdsourcing

Online crowdsourcing provides easy and economic access to human talent [15], and has been proved effective in the acquisition of corpus in various natural language processing tasks. Some work focuses on speech transcription. For example, Parent [16] proposed a two-stage approach to transcribe large amounts of speech. Lasecki et al. [17] employ non-experts to collectively caption speech in real-time to help deaf and hard of hearing people. Others [18] proposed a variety of mechanisms to collect high-quality translations for machine translation systems, and annotate text [19], [20].

In addition, crowdsourcing has been widely applied to acquisition of paraphrasing. For example, Chklovski [21] designed a game to collect paraphrases with no prompting. Negri et al. [22] designed a set of paraphrasing jobs to maximizes the lexical divergence between an original sentence and its valid paraphrases. Buzek et al. [23] proposed the idea of error-driven paraphrasing for machine translation systems. That is, they asked crowd workers to paraphrase only the parts of the input text that are problematic to the translation system. Burrows et al. [24] focused on the acquisition of passage-level samples using crowdsourcing while Lasecki et al. [25] collected dialog dataset. Recently, Jiang et al. [26] studied the key factors in crowdsourcing paraphrase collection.

The design of Challenge.AI has been inspired by many of the above approaches. However, most previous work cannot be readily applied to acquire adversarial dataset in natural language in an iterative manner.

C. The effects of prompt

When performing a task, crowd workers are influenced by instructions, examples, and context of the task [26]. Some research focuses on how different prompts can result in natural variation of human-generated language. For example, Wang et al. [27] investigated three text-based elicitation methods, e.g., sentences, scenarios, or list-based descriptions, for collecting language that corresponds to a given semantic form. Mitchell et al. [15] explored the use of crowdsourcing to generate a corpus of natural language templates for a spoken dialog system. They investigated the effect of presenting various amount of dialog content to crowd workers. Kumaran et al. [28] explored gaming as a strategy for acquisition of paraphrase data. This work presents drawing as prompt and asks the participants to produce paraphrases. Law et al. [29] examined how crowd workers are incentivized by curiosity. In this work, we investigate how prompt can be augmented by machine learning to help crowd workers generate adversarial samples.

III. Formative Study

The goal of the formative study is to understand current practice of model testing, the challenges faced by AI developers, and potential opportunities of our system.

A. Study setup

In this study, we interviewed five AI developers (denoted as D1—D5) in an IT company who are experienced in sentiment analysis. D1 is an engineer who has built sentiment classification models for different languages, such as German, English, and French. D2 is a product manager who has analyzed errors in French sentiment models. D3, D4 and D5 are research scientists who have experience in AI model design and cross-model evaluation.

We organized semi-structured interview sessions with each expert. Each interview lasted approximately 30 minutes and covered a variety of topics, starting with a general question about their experience in sentiment analysis, followed by how they test models' performance and their observation. We also focused on the challenges they encounter and how they address them. The interviewer took notes during the interviews and recorded audios for post-interview analysis. Based on the interview results, we derived four requirements to guide the design of Challenge.AI.

B. R1: Error generation

To continuously improve the performance of sentiment models, the AI developers repeat the process of *"Build (refine) model — Train model — Test model"*, where the results of model testing guide the refinement and training of models. In model testing, the AI developers (D1, D2, D3, D4) mainly rely on metrics, such as the entire accuracy of the testing dataset, the accuracy for each sentiment category, confusion matrix, F1 score, etc. One AI developer (D3) noted, *"We built sentiment analysis models for research purpose, and evaluated our models by comparing with baseline approaches on an open dataset."* However, he was not sure about the performance of their model in real-world deployment, *"If I need to deploy our model for real use, current testing would not be enough."* Since existing testing dataset is limited in coverage, D3 suggested to borrow external dataset for a comprehensive testing, *"the intuition is that, you need to increase the diversity of the testing data so as to cover different cases."* This motivates us to employ crowd force for error generation to extend the coverage of testing dataset. In addition, we should allow AI developers to collect corpus of certain category to thoroughly test the performance of models, in particular regarding corner cases.

C. R2: Error validation

After samples are crafted by crowd force, a critical task is to decide what are their "real" sentiment and whether the model makes correct predictions. High quality testing dataset is critical for evaluating the performance of a model. Some AI developers prefer human-labeled dataset because the quality is high. That motivates us to borrow the crowd to manually validate the sentiment of each generated sample. Since the sentiment is ambiguous and subjective, we plan to employ multiple crowd workers to validate one sample and use "majority vote" to mark as the ground truth.

D. R3: Error categorization

AI developers sometimes seek to obtain samples belonging to certain category to cover corner cases. For example, D2 mentioned that, "We once tested the model for biasing. We tried Asian name and western people's names to see whether

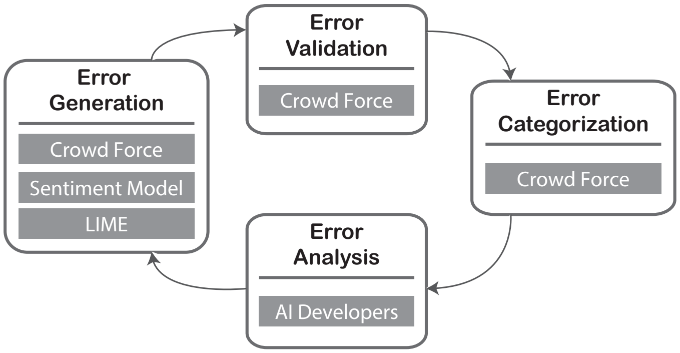
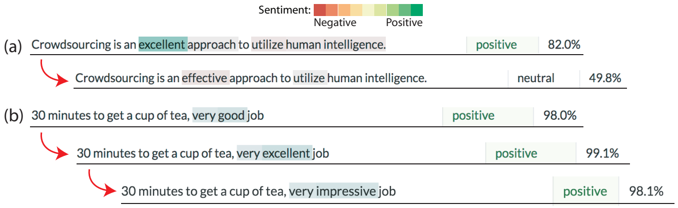


Figure 1. The architecture of Challenge.AI.



Crowdsourcing is an excellent approach to utilize human intelligence. positive 82.0%

Crowdsourcing is an effective approach to utilize human intelligence. neutral 49.8%

30 minutes to get a cup of tea, very good job positive 98.0%

30 minutes to get a cup of tea, very excellent job positive 99.1%

30 minutes to get a cup of tea, very impressive job positive 98.1%

Figure 2. The usage of LIME in two cases. (a) shows how LIME helps crowd workers modify the input sentence to successfully fool the analyzer. (b) demonstrates how LIME facilitates workers to continuously generate adversarial samples.

the model would give different predictions. We also tried like female names, male name to see if there any difference." Therefore, after obtaining samples generated by the crowd, it is critical to validate the category of them. To deal with large sample size, labeling the category using the crowd is necessary to scale the labeling process.

E. R4: Error analysis

The analysis of mis-classified samples would reveal insights to the model. However, not all samples are worth the analysis. As D2 mentioned, *"If a model makes some error predictions, they may have different impact. For example, when a sentence is negative, but the prediction is positive, that would be polarity errors. People would think the model sucks. But for sentences that are ambiguous, for example, if the ground truth is positive, the model prediction is neutral, then it would be fine. Because people can understand these errors exist."* Therefore identifying mis-classified samples with high impact would help AI developers focus on the most important errors. In addition, it would be infeasible to analyze all samples due to large sample size. As a result, demonstrating the samples at multiple levels of granularities is necessary to deal with large volume of data.

IV. CHALLENGE.AI

To understand how AI developers test models in practice, we worked closely with five AI developers in an IT company who are experienced in sentiment analysis. We conducted semi-structured interviews with them to identify the challenges they face in practice. After collecting and analyzing the interview results, we concluded the following requirements: (1) Generate errors belonging to certain category. (2) Ensure the quality of the errors. (3) Categorize errors into different groups. (4) Analyze errors to reveal insights to the model. The design of Challenge.AI is guided by these requirements. Figure 1 depicts the architecture of Challenge.AI which includes four main components, i.e., explanation-based error generation, error validation, categorization, and analysis.

A. Explanation-based error generation

This component is designed to encourage the crowd to craft sentences to fail AI models for evaluating the performance of the models. When the crowd enter the error generation component, the interface shows the introduction, example sentences belonging to a certain category, and rules of this task. After reading the instruction, a worker is able to craft a sentence in the input area. The worker then presses the "Submit" button to test the performance of the model. In response, Challenge.AI launches the sentiment analysis model in the backend, and displays the sentiment label (negative, neutral, or positive) and the probability in the result panel. The worker can verify whether the model fails or not. If it fails, the worker then needs to identify the sentiment label of the sentence.

B. Accountability via machine learning

Surrounding context may have an effect in facilitating crowd workers to craft samples, affecting the performance such as efficiency, quality, and success rate [26]. We seek to augment the prompts to assist crowd workers in error generation from two aspects, i.e., starting point and accountability, respectively. The starting point refers to the existing text in the input box, we boost the crafted sentences by providing a randomly sampled error from one category. Crowd workers are encouraged to edit the sentence in the input area.

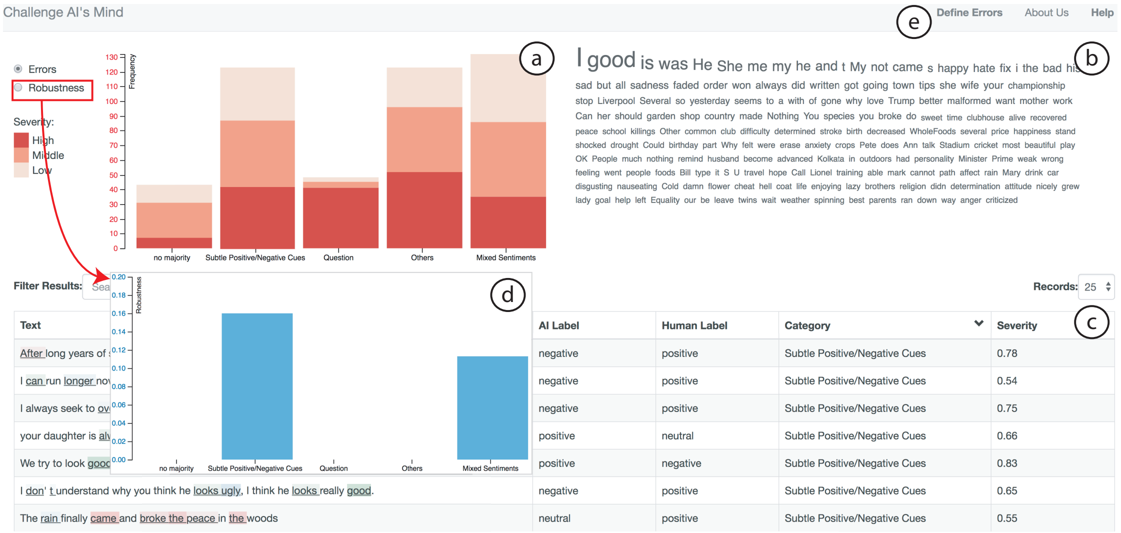
On the other hand, we provide accountability by borrowing LIME [30], an explanation technique that provides interpretable results for a prediction and is applicable to explain any models. To be specific, after a worker submits a sentence, the LIME algorithm is triggered to calculate the relationship between the prediction and each word in real time. Then the results are presented in the interface. Instead of presenting a set of numeric values, we borrow visualization techniques to intuitively depict the LIME results inline with the text. As shown in Figure 2(a), The background color of a word indicates whether it contributes to positive (green), negative (red), or neutral (yellow) sentiment.

C. Error validation and categorization

We conduct crowd-based validation and categorization by recruiting different crowd workers after the Error Generation process to obtain ground truth sentiment labels. In addition, we offer "effort-responsive" bonus to creators based on the validation results. We require at least 5 judgments for each sample, and pay $0.016 per judgment. We set up many hidden test questions for quality control, which are used to reject validations by workers who have missed a quantity of test questions [19]. The validation is performed using Figure-eight [31]

D. Error analysis

After error categorization, we obtain a dataset where each error is associated with a ground truth sentiment label validated by the crowd, a predicted label by the model, and a category.


Challenge AI's Mind
Define Errors
About Us
Help
e
a
b
c
d
● Errors
○ Robustness
Severity:
High
Middle
Low
Frequency
130
120
110
100
90
80
70
60
50
40
30
20
10
0
no majority
Subtle Positive/Negative Cues
Question
Others
Mixed Sentiments
I good is was He She me my he and t My not came s happy hate fix i the bad his
sad but all sadness faded order won always did written got going town tips she wife your championship
stop Liverpool Several so yesterday seems to a with of gone why love Trump better malformed want mother work
Can her should garden shop country made Nothing You species you broke do sweet time clubhouse alive recovered
peace school killings Other common club difficulty determined stroke birth decreased WholeFoods several price happiness stand
shocked drought Could birthday part Why felt were erase anxiety crops Pete does Ann talk Stadium cricket most beautiful play
OK People much nothing remind husband become advanced Kolkata in outdoors had personality Minister Prime weak wrong
feeling went people foods Bill type it S U travel hope Call Lionel training able mark cannot path affect rain Mary drink car
disgusting nauseating Cold damn flower cheat hell coat life enjoying lazy brothers religion didn determination attitude nicely grew
lady goal help left Equality our be leave twins wait weather spinning best parents ran down way anger criticized
Filter Results:
Sea
Records: 25
Robustness
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
no majority
Subtle Positive/Negative Cues
Question
Others
Mixed Sentiments

Text	AI Label	Human Label	Category	Severity
After long years of	negative	positive	Subtle Positive/Negative Cues	0.78
I can run longer nov	negative	positive	Subtle Positive/Negative Cues	0.54
I always seek to ov	negative	positive	Subtle Positive/Negative Cues	0.75
your daughter is alv	positive	neutral	Subtle Positive/Negative Cues	0.66
We try to look gooc	positive	negative	Subtle Positive/Negative Cues	0.83
I don' t understand why you think he looks ugly, I think he looks really good.	negative	positive	Subtle Positive/Negative Cues	0.65
The rain finally came and broke the peace in the woods	neutral	positive	Subtle Positive/Negative Cues	0.55

Figure 3. The interface allows AI developers to investigate the validated samples at different levels.

To understand the impact of each error, we define "Severity" for each error. The intuition is that, for a misclassified sentence, if both human and the model are confident about the sentiment, the mistake is severe. On the other hand, if both sides are not sure about the sentiment label, the mistake can be ignored. Hence, the severity score is calculated as $S = W_1 \times \text{Conf}_{human} + W_2 \times \text{Conf}_{AI}$, where W_1 and W_2 are weights for confidence of human and of the model, respectively. In this work, we set $W_1 = W_2 = \frac{1}{2}$. Conf_{human} represents the confidence of human, which is calculated as the percentage of the crowd making the judgment the same as majority vote. For example, for a sentence validated by five crowd workers, three of them validated the sentiment as positive. Hence, Conf_{human} becomes $\frac{3}{5} = 0.6$. Conf_{AI} is provided by the model, usually obtained as the probability or confidence of the prediction.

To help AI developers understand the model by analyzing a large quantity of errors, we build an interface to demonstrate the analysis at three different levels. After the data is loaded, the Statistic View (Figure 3(a)) uses a stacked bar chart to demonstrate the error distribution of each category at the macro-level. The x-axis presents different categories while the y-axis shows the number of errors. For each category, we manually set two thresholds to split errors into three classes representing different levels of severity, i.e., high (dark red), middle (light red), and low (pink). At the meso-level, a Cloud View (Figure 3(b)) shows a tag cloud summarizing sentiment words calculated by LIME [30]. The bigger a word is, the more frequent it appears in sentences as a sentiment word recognized by LIME. At the micro-level, a Table View (Figure 3(c)) demonstrates raw sentences, the prediction, sentiment ground truth, the category, and the severity. Various interaction techniques, such as linking and filtering, are borrowed to coordinate the three views.

V. Evaluation with the Crowd

We conducted a crowd evaluation to investigate how different prompts in error generation affect the performance of the crowd in crafting errors.

We constructed prompts based on different combination of accountability (LIME) and starting points (SP). If the starting point is empty, workers are encouraged to craft a sentence from scratch. Otherwise, workers are allowed to edit the text in the input area (SP).

TABLE I. STATISTICS OF ERROR GENERATION BASED ON TWO PROMPT CONDITIONS, e.g., A BASELINE CONDITION (NO LIME, NO SP) AND AN ENHANCED ONE (LIME, SP).

	LIME, SP	No LIME, No SP	Total
N_{total}	262	293	555
N_{valid}	75	108	183
#workers	66	46	112

We performed a between-subject design with two experimental conditions, e.g., a baseline condition (NO LIME, NO SP) and an enhanced prompt (LIME, SP), and identified two types of errors to generate, i.e., "Subtle sentiment cues" and "Mixed-sentiment" which refers to sentences containing both positive cues and negative indicators. We used Figure-eight [31] as the platform to release our error generation jobs. To be slightly generous, we paid $0.05 per sentence if the sentences successfully fail the analyzer after validation. At the same time, if the sentences belong to the required category, additional $0.05 per sentence were paid to the crowd. To reject noises and assign categories to each sample, the crowd-based validation was performed after generation.

A. Metrics

The general statistics of each job are displayed in Table I. The **Total trials**, denoted as N_{total}, include all sentences that the crowd have crafted using our system. Crowd workers have generated 249 sentences for "Subtle sentiment cues" and 306 for "Mixed-sentiment", respectively. **Validated trials** (N_{valid}) are the number of sentences that successfully fail the model based on the validation results. In addition, we count the number of distinct crowd workers for each condition (**# workers**).

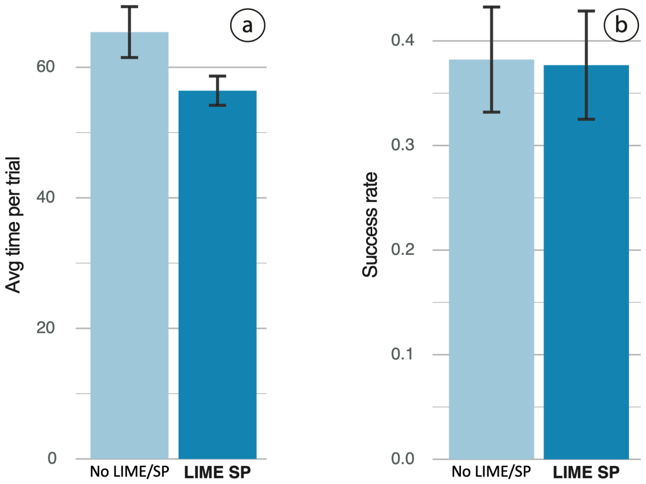


Figure 4. (a) shows the bar chart displaying average time per trial for each worker under two conditions. (b) shows how crowd workers differ in success rate. The error bars demonstrate standard errors.

Accordingly, we propose two metrics to evaluate the performance of each crowd worker. We use n instead of N to represent that the statistics values correspond to one crowd worker. **Average time per trial** (**T**) measures how much time that a worker needs to craft a trial on average, which indicates how efficient a sentence can be crafted. **Success rate** (**R_{succ}**) is measured as $\frac{n_{valid}}{n_{total}}$. This value measures how easily a worker thinks s/he can generate samples to fail the model. The success rate is useful to measure the effectiveness of prompts, as well as to analyze the vulnerability of a model.

B. Analysis of crowd performance

On average, crowd workers spent 56.4 seconds (SD=18.2) in crafting a sentence with the enhanced prompt (LIME, SP) and 65.4 seconds (SD=26.4) with (NO LIME, NO SP). Figure 4(a) shows the average time per trial (T) under each condition. We found a significant effect (t = 1.9977, p<0.05) of the enhanced prompt in reducing (T). The crowd used about 13.8% less time in crafting a sentence with (LIME, SP) than (NO LIME, NO SP). The reason may be because accountability assists workers to craft errors during the process, and editing text in the input area requires less time compared to crafting a new one from scratch. Figure 4(b) shows that crowd workers are indifferent in success rate (R_{succ}) under two conditions (38.2% V.S. 37.7%).

VI. Evaluation with AI developers

To investigate how Challenge.AI helps AI developers understand and diagnose a model, we worked with the five AI developers that we collaborated during the formative study, and organized two rounds of semi-structured interview sessions to evaluate the effectiveness and usefulness of Challenge.AI.

A. Process

We followed the architecture of Challenge.AI (Figure 1) to evaluate the entire system. Before error generation, we started from the first sessions with AI developers to obtain initial categorization for errors. Based on the category information proposed by AI developers, we used Challenge.AI to generate errors belonging to these categories, and conducted validation and categorization for crafted sentences. Finally, we organized the second interview sessions (error analysis) to understand the usefulness and limitations of Challenge.AI from the perspective of AI developers. During the entire evaluation, we used a sentiment analysis model built by D1 as the target model to test. The input of the model is a sentence, and it outputs a sentiment label associated with a probability.

1) First sessions: The goal of the first sessions is to obtain the target categories of errors to test the model. To begin with, we tested the performance of the model using a public sentiment dataset [19] where all 12284 sentences are collected from Twitter, and labeled with negative, neutral, or positive sentiment. After obtaining all misclassified sentences, we randomly sampled 200 ones and stored them in a table (CSV file format) with four columns, e.g., a 'Text' column, a 'Human_Label' column showing the ground truth, an 'AI_Label' column displaying the results calculated by the model, and an empty column titled 'Category' to allow AI developers to label a potential category for the sentence.

Each interview started with the introduction of the dataset. After that, we presented the dataset to AI developers and asked them to identify the patterns of the misclassified samples and name new categories for them. AI developers were allowed to discard sentences that are hard to be categorized. An interview took about 40 minutes. We encouraged them to express findings and thoughts using a think-aloud protocol and took notes about their feedback for further analysis.

Some AI developers have more experience in identifying patterns for errors. For example, when noticing a sentence whose benchmark label is positive, but misclassified as negative by the model, i.e., "Marissa Miller of Google makes shout out to the Khan Academy and the great things they're doing for education. #fmsignal #sxsw (cc @mention", D2 said, *"I think the model made a wrong prediction because it does not understand what 'shout out' means."* From her experience, D2 further commented that the model may not understand sentiment indications that are domain-specific or context dependent. Besides summarizing patterns in the dataset, D3 asked for sentences containing both positive and negative indicators. "Do any of them have opposite sentiment words, like, I am happy, but... something like that?" The participant further explained, *"Some models are designed to handle targeted sentiment, but determining relevant sentiment in mixed sentiment texts is challenging."* Finally we derived two categories of errors for model testing. One is called "Subtle Sentiment Cues" which means that a sentence is either positive or negative, and has positive or negative indications. The other is "Mixed-sentiment" which refers to sentences containing both positive cues and negative indicators. Further, we include three more types of errors for categorization. For example, a "Questions" category is added based on D1's comments and an "Others" is included to be more general. A "No majority" category is added after categorization if human annotators cannot reach a consensus on the category of that sample.

2) Running Challenge.AI: After obtaining the categorization, we tested the model by walking through three main components of Challenge.AI, e.g., error generation, validation, and categorization. As mentioned above, we focused on the two categories, i.e., "Subtle Sentiment Cues" and "Mixed-sentiment" in error generation while we used five categories for error categorization. The results and analysis of crowd performance are described in Section V.

3) Second interview sessions: We organized second interview sessions to evaluate how Challenge.AI helps AI developers understand the performance of the model.

After running Challenge.AI, we obtained 555 samples that 112 crowd workers generated to have successfully failed the model, where 23 errors are categorized as "Subtle Sentiment Cues" and 44 are "Mixed-sentiment". During the interviews, we demonstrated the data at three levels of granularities using the interface shown in Figure 3.

Each interview took about 45 minutes. We first presented the goal of Challenge.AI to AI developers and a detailed introduction to the data and interface. AI developers then freely explored the interface and we helped them resolve any questions they encountered. Next, the participants went through the interface to tell how they understood the performance of the model. They further identified new categories of errors by investigating detailed samples using the interface. Finally, a post-interview discussion was conducted to collect their feedback about the strengths and weaknesses of Challenge.AI. During the interview, AI developers were instructed to think aloud and we took notes about their feedback. We recorded the whole interview sessions for later analysis. We report the results of second interview sessions in the remaining of the section.

B. Value of proactive testing

A thorough testing is important for AI models before deployment. However, current practice of testing is limited in coverage, as D3 commented, *"When doing the testing, we assume that the testing dataset and training dataset are in the same feature space."* Traditional testing approach is far from enough for deploying the model in the wild, which indicates the potential value of proactive testing in evaluating the model for production. To reduce critical and embarrassing errors, AI developers are able to identify corner cases to test, and Challenge.AI collect external dataset belonging to specific categories. In addition, by investigating external dataset, AI developers can discover unseen errors. For example, our participants identified two categories that are distinct from those found in the first interview session, e.g., bias in pronouns such as 'He' and 'She', and reversed sentiment containing words like 'However', 'Though', and 'But'. Detailed discussions are reported below.

C. Getting a gist

First of all, AI developers were interested in the overall patterns of misclassified samples. The Statistics View (Figure 3(a)) provides a big picture of the entire dataset. From the stacked bar chart, D5 noticed that it is about equal distribution among high severity, middle, and low for most bars. However, the samples belonging to "Question" attracted her attention because high-severity errors account for the majority in this category. *"The model could be improved (in the 'Question' category) for sure."* D5 further explained the way of improving the model, *"In some of the supervised learning models, we need to use human heuristics to do the feature engineering (extraction) from the raw dataset. The quality of the feature extracted largely impacts the final performance."* The participant took the "Question" category as an example, *"If a model a has high probability to make severe errors for question sentences, we may specify a feature in feature engineering to detect whether a sentence is a question or a statement. So with this feature, hopefully could help the model make decisions."*

From our observation of the first sessions, all AI developers had read through about a dozen of misclassified sentences because the process of error analysis requires great mental efforts. Displaying the errors at different levels of granularities would relieve AI developers in analyzing a large number of errors. As D2 commented, *"I like the overview which gives me the impression of the entire dataset. You know, reading through two hundred errors is time-consuming and impossible (during the first interview session), and I did not do a good job last time."*

D. Examining errors by words

After examining the Statistics View, D4 switched his focus to the Cloud View showing sentiment words as tag cloud (Figure 3(b)). The participant noticed that the word "I" has the biggest font size while "Good" is the second biggest word. *"Typically in sentiment analysis, you will not expect 'I' to be particularly positive or negative. 'Good' is the second one. It makes more sense but 'I', 'is', 'was', 'he', 'me', 'my', 'she', among the first line are not sentiment words."* However, the participant changed his mind after investigating sentences containing "He" and "She". He first clicked "She" and the Table View updated. The participant noticed that the word contributes a lot to neutral sentences, and contributes once for negative and positive, respectively. Similarly, the participant further examined sentences containing the word "He", and noticed that four out of eight are negative, and "He" contributes to the negative sentiment. *"Well, it is interesting to see the difference between 'She' and 'He'. I guess the model tends to regard 'He' as a negative word."* He added, *"I think that it is necessary to examine the training data (of the model) to see whether the stop words are equal in distribution for each sentiment."*

Before using Challenge.AI, some AI developers (D1, D4, and D5) found it hard to identify patterns and categorize sentences. For example, during the first interview sessions, D4 did not know the reason for some of the predictions. The participant pointed to one question sentence and commented, *"There is no reason to label this question into negative or positive. Because it apparently contains none of the words with any sentiment."* D4 and D5 noted that they did not agree with some ground truth labels. As D4 said, *"I would recommend you have a category for mis-labeled because it is subjective."* The participant further pointed to a sentence whose benchmark label is neutral, and added, *"Now here is one, 'Social Is Too Important For Google To Screw Up A Big Launch Circus'. It sounds kind of negative to me, which is how the model classified it as."* By borrowing LIME [30] to extract sentiment words, Challenge.AI provides explanation of errors at the word level, allowing AI developers to find potential bias in the training data.

E. Reading through errors

D1 showed great interest in the exploration of samples in the "Mixed-sentiment" category. He clicked bars with dark red color under this category and read through these severe errors in the Table View. Then the participant noted, *"Some* *sentences in this category are reversed sentiment."* Then the participant pointed to a sample and added, *"Like in this case, it has the word 'but'. All content after 'but' is the content that the speaker wants to emphasize. The former part is like warm up. So the later part highlights the whole meaning of the sentence. In this case, I will not say it is a mixed sentiment. It is reversed."* Then, the participant used the search box to find all sentences containing "however" but found no sample in the table. He commented, *"I would like to test the model with sentences using reversing words, like 'but', 'however', 'although'. The model may not do a good job."*

During the first interview sessions, we realize that not all errors are worth investigation. When looking at the errors, D5 commented, *"A lot of these are difficult for human. For those which are less obvious, you may ask three different people and got three difficult answers."* The participant further added, *"Since sentiment analysis is subjective, if an error is ambiguous to human, I do not think the model made a severe mistake."* Therefore, the definition of severity helps AI developers focus on errors that are important to examine.

VII. Design Implications

Proactive testing is a promising direction that helps AI developers get more insights into the model. Challenge.AI is the first prototype that supports proactive testing using the crowd force, and we suggest the following aspects that future research can explore.

First, *include all the generated data by the crowd including those that can fail the model and those cannot.* Because only the misclassified samples are not enough to help AI developers understand how the model performs in some cases. For example, D2 has found two sentences containing the word "Trump" by filtering. However, the participant could not conclude whether the model is biased to the word "Trump". D2 commented, *"I am only looking at the errors. It is hard to tell (whether the model is biased to "Trump"). I mean, these errors could be 99% of the instances in which case the model is doing very poorly. But this could be less than 1% of the instances in which case the model is doing fantastic."*

Second, *apply better explanation techniques.* In this study, we choose the LIME algorithm [30] to identify and highlight sentiment words related to the prediction. However, our participants found that some sentiment words are confusing. For example, D4 found a positive sentence with AI labeled negative, "I can run longer now". The word "can" is highlighted in green (positive) and "longer" highlighted in blue (neutral). He commented, *"The AI label is negative. However, it is wired that no words are marked as negative."* However, when more advanced analytical techniques are developed in the future, such issue may be resolved.

Third, *enhance the generation component for word-level categories.* Challenge.AI has been proved to be effective in collecting samples belonging to *concept-level* categories such as "mixed-sentiment" and "subtle sentiment cues". However, AI developers may sometimes seek to test the model using samples containing certain words, such as "Trump". Intuitively, collecting samples with certain words could be more cost- and time-efficient by using techniques in information retrieval. We plan to study how various information retrieval techniques help in collecting samples of different category.

Fourth, *provide real-time feedback for proactive testing.* The main process of sample collection, e.g., generation, validation, and categorization, takes a long time and AI developers cannot test the model in real-time. One possible solution is to borrow workflows from real-time crowdsourcing [32], [33], [34], [35] to reduce the delay in obtaining the testing results. Another solution is to augment the error analysis interface as suggested by D2, *"Since the model is already trained. Maybe you can (embed the model in the backend and) add an input box for real-time testing so that I can test some of the sentences in my mind."*

Fifth, *augment error analysis with advanced analytical methods.* our system borrows knowledge from AI developers to identify new patterns to test. However, the process is time-consuming and not scalable. It would be beneficial to incorporate automatic analytical methods, such as text classification or clustering, to assist AI developers in summarizing patterns among errors.

VIII. Conclusion and Future Work

To summarize, we propose Challenge.AI, a crowd system that supports proactive testing for AI models by extending the coverage of testing dataset with crowd-generated errors. To assist error generation, we propose an explanation-based error generation technique combining human intelligence and machine learning. We use crowd evaluation to compare the explanation-based error generation technique and a baseline approach. In the future, we plan to establish metrics to compare the generated dataset and open sourced ones from different perspectives, such as the topic coverage, syntactic structure, and uni-gram distribution, to have a comprehensive understanding of the crowd-crafted dataset.

References

[1] https://www.forbes.com/sites/parmyolson/2018/02/26/artificial-intelligence-ai-bias-google/#5e6155b31a01, retrieved: 2020-02-20.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[3] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and music adversaries," IEEE Transactions on Multimedia, vol. 17, no. 11, 2015, pp. 2059–2071.

[4] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," arXiv preprint arXiv:1607.02533, 2016.

[5] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," arXiv preprint arXiv:1605.07277, 2016.

[6] N. Papernot and et al., "Practical black-box attacks against machine learning," in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 2017, pp. 506–519.

[7] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," arXiv preprint arXiv:1710.11342, 2017.

[8] B. Liang and et al., "Deep text classification can be fooled," arXiv preprint arXiv:1704.08006, 2017.

[9] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," arXiv preprint arXiv:1612.08220, 2016.

[10] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," arXiv preprint arXiv:1801.04354, 2018.

[11] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, 2018, pp. 31–36.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in AAAI Conference on Artificial Intelligence, 2018.

[13] N. Papernot, P. McDaniel, A. Swami, and R. Harang, “Crafting adversarial input sequences for recurrent neural networks,” in Military Communications Conference, MILCOM 2016-2016 IEEE. IEEE, 2016, pp. 49–54.

[14] S. Samanta and S. Mehta, “Towards crafting text adversarial samples,” arXiv preprint arXiv:1707.02812, 2017.

[15] M. Mitchell, D. Bohus, and E. Kamar, “Crowdsourcing language generation templates for dialogue systems,” Proceedings of the INLG and SIGDIAL 2014 Joint Session, 2014, pp. 172–180.

[16] G. Parent and M. Eskenazi, “Toward better crowdsourced transcription: Transcription of a year of the let’s go bus information system data,” in 2010 IEEE Spoken Language Technology Workshop, 2010, pp. 312–317.

[17] W. Lasecki and et al., “Real-time captioning by groups of non-experts,” in Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, ser. UIST ’12. New York, NY, USA: ACM, 2012, pp. 23–34. [Online]. Available: http://doi.acm.org/10.1145/2380116.2380122

[18] O. F. Zaidan and C. Callison-Burch, “Crowdsourcing translation: Professional quality from non-professionals,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, ser. HLT ’11. Association for Computational Linguistics, 2011, pp. 1220–1229. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002626

[19] 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 (SemEval-2017), 2017, pp. 502–518.

[20] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, “Semeval-2016 task 4: Sentiment analysis in twitter,” in Proceedings of the 10th international workshop on semantic evaluation (semeval-2016), 2016, pp. 1–18.

[21] T. Chklovski, “Collecting paraphrase corpora from volunteer contributors,” in Proceedings of the 3rd International Conference on Knowledge Capture, ser. K-CAP ’05. ACM, 2005, pp. 115–120. [Online]. Available: http://doi.acm.org/10.1145/1088622.1088644

[22] M. Negri, Y. Mehdad, A. Marchetti, D. Giampiccolo, and L. Bentivogli, “Chinese whispers: Cooperative paraphrase acquisition.” in LREC, 2012, pp. 2659–2665.

[23] O. Buzek, P. Resnik, and B. B. Bederson, “Error driven paraphrase annotation using mechanical turk,” in Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk. Association for Computational Linguistics, 2010, pp. 217–221.

[24] S. Burrows, M. Potthast, and B. Stein, “Paraphrase acquisition via crowdsourcing and machine learning,” ACM Trans. Intell. Syst. Technol., vol. 4, no. 3, 2013, pp. 43:1–43:21. [Online]. Available: http://doi.acm.org/10.1145/2483669.2483676

[25] W. S. Lasecki, E. Kamar, and D. Bohus, “Conversations in the crowd: Collecting data for task-oriented dialog learning,” in First AAAI Conference on Human Computation and Crowdsourcing, 2013.

[26] Y. Jiang, J. K. Kummerfeld, and W. S. Lasecki, “Understanding task design trade-offs in crowdsourced paraphrase collection,” arXiv preprint arXiv:1704.05753, 2017.

[27] W. Y. Wang, D. Bohus, E. Kamar, and E. Horvitz, “Crowdsourcing the acquisition of natural language corpora: Methods and observations,” in 2012 IEEE Spoken Language Technology Workshop (SLT), 2012, pp. 73–78.

[28] A. Kumaran, M. Densmore, and S. Kumar, “Online gaming for crowdsourcing phrase-equivalents,” in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 1238–1247.

[29] E. Law, M. Yin, J. Goh, K. Chen, M. A. Terry, and K. Z. Gajos, “Curiosity killed the cat, but makes crowdwork better,” in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ser. CHI ’16. New York, NY, USA: ACM, 2016, pp. 4098–4110. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858144

[30] M. T. Ribeiro, S. Singh, and C. Guestrin, “"why should I trust you?": Explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, 2016, pp. 1135–1144.

[31] https://www.figure-eight.com/, retrieved: 2020-02-20.

[32] W. S. Lasecki, K. I. Murray, S. White, R. C. Miller, and J. P. Bigham, “Real-time crowd control of existing interfaces,” in Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, ser. UIST ’11. ACM, 2011, pp. 23–32. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047200

[33] G. V. de la Cruz, B. Peng, W. S. Lasecki, and M. E. Taylor, “Towards integrating real-time crowd advice with reinforcement learning,” in Proceedings of the 20th International Conference on Intelligent User Interfaces Companion, ser. IUI Companion ’15. New York, NY, USA: ACM, 2015, pp. 17–20. [Online]. Available: http://doi.acm.org/10.1145/2732158.2732180

[34] A. Lundgard, Y. Yang, M. L. Foster, and W. S. Lasecki, “Bolt: Instantaneous crowdsourcing via just-in-time training,” in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ser. CHI ’18. New York, NY, USA: ACM, 2018, pp. 467:1–467:7. [Online]. Available: http://doi.acm.org/10.1145/3173574.3174041

[35] Z. Liao, Y. Xian, X. Yang, Q. Zhao, C. Zhang, and J. Li, “Tscset: A crowdsourced time-sync comment dataset for exploration of user experience improvement,” in 23rd International Conference on Intelligent User Interfaces, ser. IUI ’18. New York, NY, USA: ACM, 2018, pp. 641–652. [Online]. Available: http://doi.acm.org/10.1145/3172944.3172966

Inter-operating Co-opeting Entities

A Peer-to-Peer Approach to Cooperation between Competitors

Alain Sandoz
Informatics Institute,
University of Neuchâtel
Neuchâtel, Switzerland
e-mail: alain.sandoz@unine.ch

***Abstract*—The paper identifies conditions that enable competing business entities to interoperate through their mutual cooperation function while maintaining a strict separation of their competing functions, and in particular the secure operations of their core IT-business infrastructure. Implications on the architecture of the cooperation function and an implementation realized using the Kubernetes micro-service infrastructure and Hyperledger Fabric are presented.**

Keywords - horizontal cooperation; co-opetition; peer-to-peer network; micro-service architecture; Kubernetes; Hyperledger Fabric.

I. Introduction

This short paper describes work in progress started in 2017 that has led our team to design and implement generic mechanisms to enhance interoperability among distrustful actors. The resulting *peer-to-peer* configuration enables private entities (i.e., competitors) and/or public entities (i.e., regulators) to interoperate under good conditions where they are willing or compelled to collaborate. On the other hand, functions that are too sensitive to be exposed to information leakage or tampering from an entity's environment are kept safe (up to the level of safety provided by each entity for its own resources).

The initial project was meant to speed up and improve information flow between the many actors of the milk production and processing sector in Switzerland: farmers, transporters, label organizations, laboratories, the dairy industry (buyers, transformers, retailers, exporters), regulators, and of course the end-consumer, all require information in a maze of formats and temporalities. Often, the complexity of data-management and the lack of synchrony between data-flows and the actual logistics of production along the value chain prevent improvements or paralyze processes. Even competing entities were willing to work together to overcome difficulties, i.e., cooperate.

Our work on behalf of the milk sector delivered mechanisms that apply to other sectors of the economy or of society, including, e.g., banking, insurance, and healthcare.

The results we describe pertain to specific business conditions called co-opetition, together with specific technical conditions that are found in distributed systems, in particular, but not restricted to peer-to-peer (P2P) networks.

The paper is structured as follows. Section II briefly describes the state of the art from where we start. In section III we define the *cooperation function* and the *coordination function* in the context of interoperability. In section IV we state a small set of conditions, or *principles*, for the digitalization, integration, and interoperation of the cooperation function by and among co-opeting entities and define the architecture of a P2P network that operates the cooperation function. In section V we describe a productive implementation of the concept. We conclude in section VI with implications on co-opetition among software-providers and a possible transformation of some regulatory tasks currently implemented *de facto* in the form of centralized coordination functions among networks of competing economic actors.

II. State of the Art

This section briefly describes the notions of co-opetition, peer-to-peer collaboration, and cooperation function.

A. Horizontal cooperation, co-opetition

In business, many situations arise where competitors must cooperate to sustain their access to the market, reduce costs, or collectively realize positive conditions that would be impossible on an individual basis. This was first described in [1]. It happens e.g., in logistics and transport [2], in industry [3], in banking [4], and is generally called "horizontal cooperation" or co-opetition [5], as opposed to the master-slave-type of dependency between customers and suppliers in a vertical setting, or the possible cooperation of business entities that are not competitors.

Co-opetition is a sensitive endeavor, where cooperation between competitors on some specific function is *beneficial*, whereas the core business goal of each party in the cooperation remains *domination* of the other(s).

Since cooperation implies the sharing of resources, e.g., at least of information, to manage which resources can be shared for mutual benefit without compromising individual survival is delicate.

B. Peer-to-peer networks

On the other hand, peer-to-peer collaboration in computer networks is a well-established practice for groups of otherwise autonomous entities to share commonly valuable resources [6], [7]. Shared resources may be files,

ISBN: 978-1-61208-785-6

computing power, voice over internet protocol (VoIP) [8], partial solutions to broken-down problems, storage, etc.

An interesting aspect of P2P networks is that parties to the P2P activity usually have no additional interaction. Their common interest and benefit can be to swap multimedia resources, solve together a genome or signal-processing problem, take part in an elaborate mesh-based resource sharing configuration [9], participate in multiplayer games [10], or even provide "services" as in [11], but in general, stakeholders in P2P activities do not otherwise interact.

The absence of "external relationship" is not necessary in P2P networking, nor does the latter *a priori* exclude competition (in the business sense) between peers.

Though competition between stakeholders in a peer-to-peer network is not excluded, it does require some caution: P2P activities imply that peers execute foreign operations on their computer infrastructure, or, to be more precise, some computer infrastructure that they control. Therefore, certain conditions should be guaranteed to stakeholders before they open-up their strategic resources, information, processes, or core-functions like customer relationship management systems (CRM) to "peers" that are, in essence their enemies.

C. Cooperation and coordination functions

The functions shared between co-opetitors are called *cooperation* functions and *coordination* functions. They are usually implemented on the IT infrastructure of a central actor, e.g., SWIFT [4]. For many reasons including trust and fault-tolerance, centralization of any function should however be avoided in P2P networks, the more so if peers are inherently distrustful of each other.

The approach we follow is *fully distributed*: there is no central component and every peer operates and executes every function that it requires on its own. Execution of functions is traced and logged, and *correct behavior* can *always be proven* unless the conditions for consensus (among misbehaving peers) are met and used to disqualify an honest peer. This is about the best that one can reach under the general conditions of distributed computing systems.

III. COMPETITION, COOPERATION AND COORDINATION

Two important features of co-opeting entities in the traditional business environment are [12]: *1)* the separation of the cooperation function from the core business functions within each of the co-opeting entities, and *2)* the presence in some form of a coordinating actor.

The first feature is easily understandable for the security of the core function of the business, but is also related to the different social and relational skills of "competitor-" and "cooperator-" types of workers in any given company.

Figure 1 illustrates the situation of a set of competing entities CE-***1***, CE-***2***, … CE-***N***, that compete to access their share of the market, whereas they each operate a cooperation function clearly separated from the core functions of the business with defined access conditions. A *coordination function* is necessary to establish consistency among entities with regards to cooperation. Of course, cooperation among entities in complex business environments like global transport or banking requires a computer-supported coordination function. As in [4], this function is often complex itself and is centralized, which enables strong semantics of transactions (e.g., non-repudiation of bank transfers in SWIFT). In this case, the coordination function might be owned and operated, or at least controlled by the community of co-opetitors.

A central exogenous component of this type implies a (very heavy) client-server model of coordination, as opposed to the fully distributed model we are looking after.

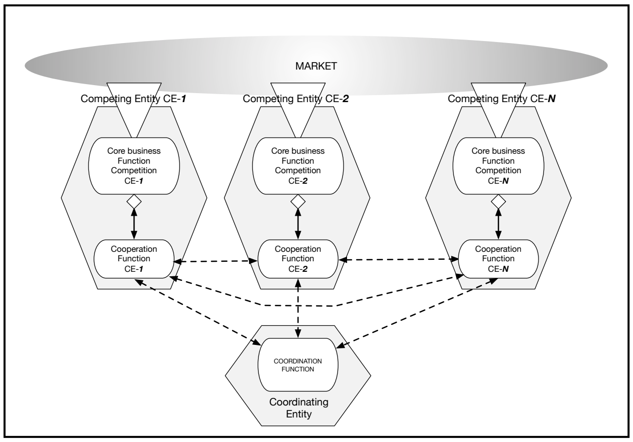

MARKET
Competing Entity CE-1
Competing Entity CE-2
Competing Entity CE-N
Core business Function Competition CE-1
Core business Function Competition CE-2
Core business Function Competition CE-N
Cooperation Function CE-1
Cooperation Function CE-2
Cooperation Function CE-N
COORDINATION FUNCTION
Coordinating Entity

Figure 1. Business entities that compete *and* cooperate.

Huge sets of co-opeting entities like SWIFT or large stock-exchange platforms might need to manage billions of transactions monthly (that incidentally generate millions of euros of costs for the stakeholders) with a central coordination function, or they might be too big to get rid of a system that was once a solution to their reliability problem.

However, co-opetition on a smaller scale doesn't require and usually cannot economically support, a centralized, dedicated and humanly operated coordination function. In this case, the coordination function might consist only in managing reliable communication and consensus on a small set of global state values necessary for all actors to make mutually consistent and locally secure decisions.

Figure 2 illustrates this situation: the cooperation functions of co-opeting entities interact within a P2P network. Access by the coordination function to the core IT infrastructure of each competing entity is strictly controlled. There is no more active autonomous coordination function.

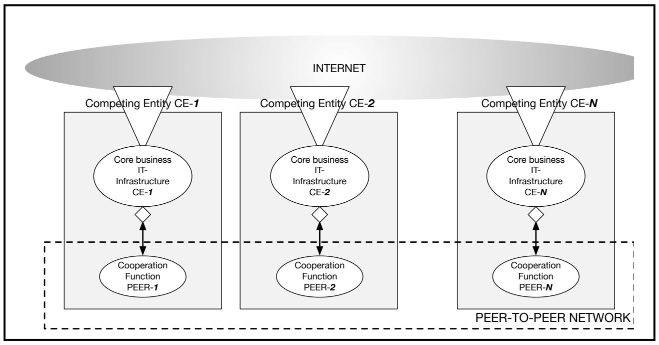

INTERNET
Competing Entity CE-1
Competing Entity CE-2
Competing Entity CE-N
Core business IT-Infrastructure CE-1
Core business IT-Infrastructure CE-2
Core business IT-Infrastructure CE-N
Cooperation Function PEER-1
Cooperation Function PEER-2
Cooperation Function PEER-N
PEER-TO-PEER NETWORK

Figure 2. To implement cooperation in a peer-to-peer network.

In this representation, each competing entity is a *node* of the peer-to-peer network. At first glance, it should be possible to implement this configuration with traditional web-services (e.g., SOAP/XML or REST). However, this is not the case, specifically because a distributed coordination function cannot consistently be implemented using web-services that operate between IT-infrastructures of independent competing entities. Also, note that using a distributed database system in this situation is technically equivalent to using a centralized coordination function.

IV. Architecture of the Peer-to-Peer Network

In order for the cooperation- and the distributed coordination- functions to be implemented within a set of co-opeting entities the following conditions are necessary:

1) each competing entity completely, autonomously, and separately operates its own core functions, **and** each competing entity operates an instance of the cooperation function within a node of the P2P network;

2) each instance of the cooperation function maintains state values of the global coordination function. State values can be local values of the cooperation function instance executed on some node; or distributed state values that are consistent among the cooperation function instances of a subset of competing nodes; or consensus state values that must be kept consistent on all running cooperation function instances of the set of competing nodes;

3) the set of all coordination function instances of the group of competing entities, together with all state values of all types of these instances, defines a consistent distributed information system under conditions 1) and 2) above.

Since the cooperation function is specific to the type of business activity considered, the architecture of the P2P network and of each node have to provide some domain-specific services (in particular, the services required by the coordination function, if any), whereas the conditions that pertain to the operation of a distributed system rely on general-purpose services. This is illustrated in Figure 3.

Since the IT-infrastructures operated by different competing entities are by nature different, the implementation of the cooperation function, i.e., the implementation of the individual nodes in the P2P network, are bound to be different. The manner by which competing entities are brought to trust each other with regards to the correctness of their competitors' cooperation function (and possibly the underlying coordination function) can be left to each group of entities. However, *certification of nodes*, *fully traced communication*, and *non-repudiation* (in the sense that correct behavior of a peer can always be proven) are properties that can help foster trust. These features were implemented in the project described in the next section.

V. Implementation

To illustrate the development above, we briefly present the implementation of the peer-to-peer network that was implemented in view of [13]. In this case, the cooperation function was relatively complex (managing the transmission, authorized by their owners, of information between operators of public or private databases of farm-related data) with a coordination function that enabled data-owners (i.e., farmers) to enforce in real time together with each competing entity concerned, who was entitled to receive their data.

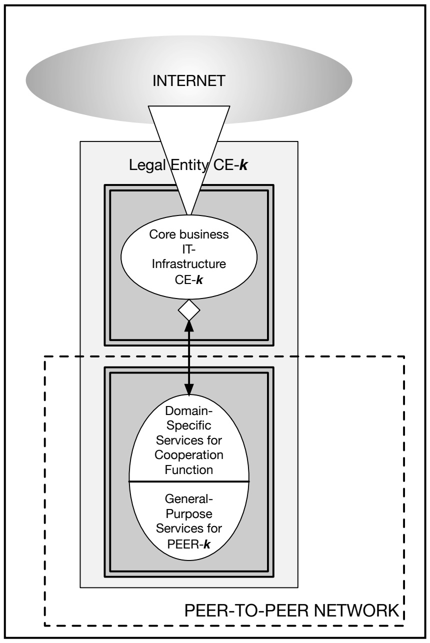

INTERNET
Legal Entity CE-*k*
Core business IT-Infrastructure CE-*k*
Domain-Specific Services for Cooperation Function
General-Purpose Services for PEER-*k*
PEER-TO-PEER NETWORK

Figure 3. Architecture of a peer (node).

Because of the nature of the competing entities (often small, autonomous, and distrustful organizations with weak or external IT-resources), it was decided to provide the cooperation function, with each competing entity's node, in a separately operable Kubernetes (K8s) [14] cluster (see Figure 4). The goal is to facilitate integration and long-term maintenance by using standard infrastructure components.

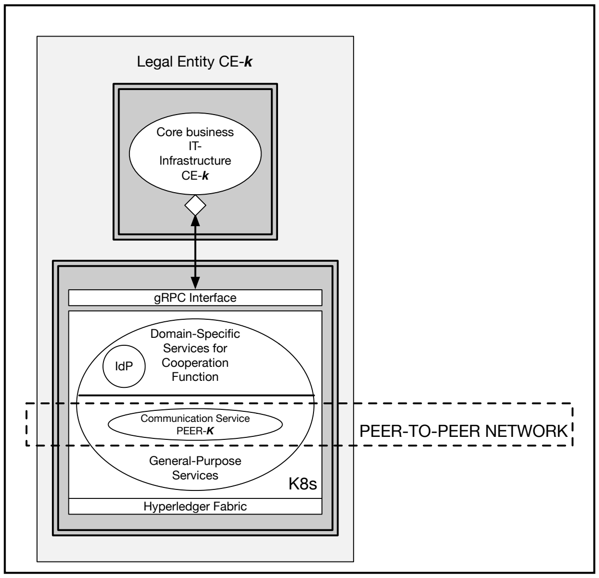

Legal Entity CE-*k*
Core business IT-Infrastructure CE-*k*
gRPC Interface
Domain-Specific Services for Cooperation Function
IdP
Communication Service PEER-*K*
PEER-TO-PEER NETWORK
General-Purpose Services
K8s
Hyperledger Fabric

Figure 4. Implementation of a node with gRPC / K8s / HLF.

This feature also enables competing entities to outsource the construction and operation of their nodes, and at the same time withhold legal control and responsibility on their operations with contracts and service level agreements.

The interface (API) between the core function and the cooperation function is realized using the gRPC [15] framework. Identity and access management is realized with OpenID Connect/OAuth 2.0 [16]. The coordination function will rely on Hyperledger Fabric [17] (which has a K8s implementation) and is currently only partially realized with the ledger (integration, along with the implementation of traceability for sensitive products or objects in the value-chain using the ledger are planned in future work).

Hyperledger Fabric is a permissioned ledger well adapted to the situation of a set of co-opeting entities (by nature reconfigurable at any time, but not dynamic in real time nor open to unidentified / unqualified peers).

The implementation is meant for private and for public entities alike, in the agro-food sector. The heterogeneity of actors tolerates a model infrastructure like Kubernetes (that peers can bypass using another implementation at their own risk), and the usage of a permissioned ledger for storing and accessing global state values; it does not however allow the general usage of a blockchain for the storage of local or transactional state values, because of the IT policy of each peer that could possibly prevent it, e.g., for public agencies.

VI. Conclusion

We have shown how the local cooperation- and the distributed coordination- functions of a group of co-opeting entities can be implemented in a peer-to-peer network.

The successful implementation of the approach leads to two remarks. First, *co-opetition in business requires some sort of co-opetition among the software-providers of the business entities concerned.* Lines of business in economic sectors have their established sets of IT-tools and SW-providers (e.g., SAP among others for the enterprise resource planning core function of a business). If a group of business entities is lead to co-opete in its sector, then the group of associated software-providers should do so also: in order to supply their customers with the necessary cooperation and coordination functions (i.e., cooperate) and thus remain competitive on that market.

Second, if some economic activity is subject to central regulatory coordination (control) as in banking or in animal production, then *each business entity that is registered for the activity must implement and operate the coordination / control function in its processes and its IT*.

The approach proposed in this paper shows how the coordination / control function of the regulator could be distributed among these co-opeting entities in a P2P framework under good conditions (i.e., using the group of entities concerned to overlook the correct implementation of the coordination / control function). This could lead in some sectors to replace costly and rigid control structures embedded in public administrations by distributed resources that already operate the same functions, possibly reducing by half the cost of some regulatory controls.

These remarks, as well as the implementation of the ledger as a means for traceability in the cooperation function are the subject of ongoing and future work.

Acknowledgment

This work was conducted with the help and precious contribution in Switzerland of Léa Stiefel, STS-Lab, University of Lausanne (lea.stiefel@unil.ch), of Alain Buehler, Yves Gabry, and Jordan Latinov, and finally of Sunny Soft Ltd. in Sofia, Bulgaria.

References

[1] B. Nalebuff and A. Brandenburger, Co-opetition, paperback ed., London: Profile Books Ltd, 2002 (first published 1996)

[2] F. Cruijssen, W. Dullaert, and H. Fleurens, “Horizontal Cooperation in Transport and Logistics: A Literature Review”, Transportation Journal, Vol. 46, No. 3, pp. 22-39, Summer 2007, doi: 10.2307/20713677

[3] A. Kosansky and T. Schaefer ,“Should you swap commodities with your competitors?”, CSCMP Supply Chain Quarterly, 2010, https://www.supplychainquarterly.com/topics/Logistics/scq201002swap/

[4] Society for Worldwide Interbank Financial Telecommunication (SWIFT) [retrieved: Feb. 2020] from https://en.wikipedia.org

[5] M. Bengtsson and S. Kock, “ “Coopetition” in Business Networks—to Cooperate and Compete Simultaneously”, Industrial Marketing Management, vol. 29, pp. 411–426, Sept. 2000, doi: 10.1016/S0019-8501(99)00067-X

[6] S. Androutsellis-Theotokis and D. Spinellis, “A Survey of Peer-to-Peer Content Distribution Technologies”, ACM Computing Surveys, Vol. 36, No. 4, Dec. 2004, pp. 335–371, doi: 10.1145/1041680.1041681

[7] R. Sarkar, “Distributed Systems - Peer-to-Peer”, University of Edinburgh, Lecture Notes, Fall 2014, [retrieved: Feb. 2020] from https://www.inf.ed.ac.uk/teaching/courses/ds/slides1415/p2p.pdf

[8] F. Victora Hecht and B. Stiller, “Enabling Next Generation Peer-to-Peer Services”, Proceedings of the 2nd international conference on Autonomous Infrastructure, Management and Security: Resilient Networks and Services, Jul. 2008, pp. 211–215

[9] C. Canali, M.E. Renda, P. Santi, and S. Burresi, “Enabling Efficient Peer-to-Peer Resource Sharing in Wireless Mesh Networks”, IEEE Transactions on Mobile Computing, March 2010, doi: 10.1109/TMC.2009.134

[10] M. Boron,·J. Brzezinski, and A. Kobusinska, “P2P matchmaking solution for online games”, Peer-to-Peer Networking and Applications, Jan. 2019, doi: 10.1007/s12083-019-00725-3

[11] J. Gerke, D. Hausheer, J. Mischke, and B. Stiller, “An Architecture for a Service Oriented Peer-to-Peer System (SOPPS)”, Praxis der Informationsverarbeitung und Kommunikation, June 2003, doi: 10.1515/PIKO.2003.90

[12] R. Leitner, F. Meizer, M. Prochazka, and W. Sihn, “Structural concepts for horizontal cooperation to increase efficiency in logistics”, CIRP Journal of Manufacturing Science and Technology, vol. 4, pp. 332–337, 2011, doi: 10.1016/j.cirpj.2011.01.009

[13] L. Stiefel and A. Sandoz: “Reshaping Swiss Agriculture Through A Peer-To-Peer Approach”, Society for Social Studies of Science, Annual Meeting, New Orleans, Sept. 2019

[14] Kubernetes, [retrieved: Feb. 2020] from: https://en.wikipedia.org/wiki/Kubernetes

[15] gRPC, [retrieved: Feb. 2020] from: https://grpc.io

[16] OpenID, [retrieved: Feb. 2020] from: https://openid.net/connect/

[17] Hyperledger Fabric, [retrieved: Feb. 2020] from: https://en.wikipedia.org/wiki/Hyperledger

Using Recurrent Neural Networks to Predict Future Events in a Case with Application to Cyber Security

Stephen Jacob
Dept. of Electronics and Informatics
Athlone Institute of Technology
Athlone, Ireland
Email: s.jacob@research.ait.ie

Yuansong Qiao
Dept. of Electronics and Informatics
Athlone Institute of Technology
Athlone, Ireland
Email: ysqiao@research.ait.ie

Paul Jacob
Dept. of Electronics and Informatics
Athlone Institute of Technology
Athlone, Ireland
Email: pjacob@ait.ie

Brian Lee
Dept. of Electronics and Informatics
Athlone Institute of Technology
Athlone, Ireland
Email: blee@ait.ie

***Abstract*—Due to the number of cyber attacks targeting business organisations daily, anomaly detection software generates large numbers of alerts. While this information is invaluable to Incident Response Teams, one problem is to prioritize these alerts and to distinguish between those that signal a serious threat to network enterprises and low priority alerts. One approach is to use a model that relates an organisation's missions, processes, services and infrastructure. By predicting future events in existing business processes, and subsequently using this model to identify associated services and infrastructure, cyber security personnel can prioritize critical alerts that threaten these assets. Long Short Term Memory based deep learning models are suited to modeling sequential data, and in particular can model long term dependencies in sequences. This paper evaluates the use of such models to predict subsequent events in ongoing cases. Two training techniques are applied to four data sets. The techniques are evaluated with respect to the accuracy of the predictions and their performance on predicting frequent and infrequent events.**

Keywords*-*Process Mining; Deep Learning; Recurrent Neural Networks; LSTM; Cyber Security.

I. Introduction

Most business organisations are constantly targeted by cyber security attacks. Anomaly detection software generates huge volumes of alerts and Computer Security Incidence Response Teams (CSIRT) struggle to follow up on all of these alerts. They require means of distinguishing alerts that signal attacks on critical business processes from low-priority alerts [1].

One way to do this is to predict future events in currently executing business processes and with the aid of a mission dependency model as outlined in Section II, identify critical services and infrastructure in the organisation. Security alerts, which target these critical services and assets can then be prioritized for the attention of the CSIRTs.

The main aim of this paper is to investigate the application of deep learning to process mining as a means to indicate likely high priority security events. The objective is to use Recurrent Neural Networks (RNN), in particular Long Short Term Memory (LSTM) networks, to model event traces with a view to using the resulting model to predict future events. The use of process mining for cyber security attack and anomaly detection has been demonstrated by the work of Mauser et al. [2], Alvarenga et al. [3] and van der Aalst et al. [4]. Process mining techniques have also been used for visualisation of cyber attacks [1]. Recent research undertaken by Tax et al. [5] and Evermann et al. [6] highlights that using deep learning applications to model and predict process sequences is an effective and increasingly popular approach.

In this paper, we investigate two methods to train LSTM networks, which model ongoing business processes. We evaluate both methods and determine which one is the more effective at training an LSTM network to predict subsequent events. The first method generates prefixes from every sequence in the data and trains the network to predict the next event from these prefixes. The second method, Teacher Forcing [7], trains the network at every time step as a case/sequence of events is passed through the network. Both these methods are applied to four data sets, the Business Process Intelligence (BPI) challenges from [8]–[10], and the Helpdesk data set used as supplementary material for Tax et al. [5].

The paper is structured as follows: Section II explores previous related approaches to the use of deep learning to monitor process sequences. Section III provides relevant background information in Mission Dependency Modeling, Process Mining and RNN/LSTM Neural Networks. Section IV describes the two approaches to training a LSTM network mentioned above. Section V outlines the experimental setup and evaluation. Section VI presents the results obtained from the experiment. Section VII is the conclusion.

II. Related Work

Alvarenga et al., [1] addressed the concept of alert correlation, as well as the issue that an Intrusion Detection System (IDS) produces an unmanageable amount of alerts and most are low-level annoying alerts incorrectly categorized as malicious. The overwhelming number of alerts results in keeping network administrators from responding appropriately to the more critical attack forms used by cyber attackers. Alvarenga proposed a process mining method to produce process models to assist administrators to identify and investigate multistage cyber attacks.

Alvarenga et al., [3] carried out a study to discover cyber attack strategies targeting networks using traditional process mining and the open-source process mining framework ProM [11]. A data log is generated and taken from an IDS deployed at University of Maryland. The data was loaded into the ProM framework and a process model was generated to visualize the process paths extracted from the data set. Further analysis identified the causal dependencies between events by comparing different cases of sequential events in the model. A benefit of this was that the discovered model could be used to visualize alerts consistent with that of a cyber attacker's perspective when attempting to compromise a targeted network.

Mauser et al., [2] used process mining discovery to detect and identify cyber security attacks on enterprise systems. Mauser identified cyber attacks by detecting anomalies in process executions in a software system. By visualizing said execution paths using Petri Nets, irregular processes paths could be isolated by comparing them to a model of regular process activity.

Tax et al., [5] applied LSTM networks to the BPI 2012 and Helpdesk data sets to learn from predicting both the subsequent event and the time until the next event. They evaluate a number of different neural network architectures with two or more network layers ranging from completely separate networks to predict activity and time to various combinations of shared and specific layers for both predictions. The method used in this paper to train the neural network is the prefix method. We re-implement this method and also implement the alternative teacher forcing method, which results in substantial improvements in training times.

Evermann et al., [6] also used an LSTM network to predict future events in a case. Evermann's approach is motivated by identifying the associated resources for an event and detecting the long-lasting dependencies within cases to subsequently predict future events. Associated resources for an event include the duration of an event, and the related resources (personnel, attributes) assigned to them. Evermann applied this approach to the BPI 2012 and 2013 data sets [8][9]. In addition, we make predictions for the BPI 2014 data set [10]. To the best of our knowledge, event prediction has not been previously applied to this data set.

III. Background Information

In this section, we first describe the mission dependency metamodel we use in our machine learning approach. We then provide an overview of the technologies used: process mining, neural networks, recurrent neural networks and LSTM models.

A. Mission Dependency Metamodel

Mission dependency modelling is a technique used as part of cyber risk assessment. This model makes explicit the relationships between mission objectives, business processes, IT services and computing assets of an organisation.

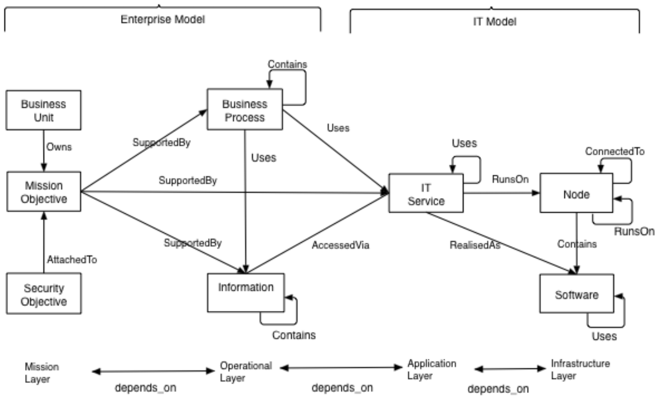


Figure 1. Mission Dependency Metamodel.

One such approach is the dependency model shown in Figure 1 that was introduced in [12]. The four layers in the model are the Mission, Operation, Application and the Infrastructure Layers. We are focused on the Operation Layer and are interested in predicting future events in currently executing business processes. This information when mapped through the dependency graph to the underlying layers can be used to identify critical services and infrastructure that may be liable to attack in the short term arising from the current cyber security situation. This, in turn, can help security response teams to prioritize security alerts in such a way as to best protect critical processes in an organisation.

B. Process Mining

Process mining can be collectively defined as the analysis, discovery and modeling of information extracted from process data sets [2][13]. These data sets are comprised of cases, which are process execution paths or sequences of events. Traditional business process mining can discover process models from event data using, e.g., the Alpha algorithm [14]. Graphical models can be generated and observed using a variety of tools including the open-source framework ProM. Once a model exists, conformance checking can be carried out to determine if logged cases conform to the model. Other insights include the ability to audit and analyze the data process, as well as how to improve it.

C. Neural Networks

An alternative to the traditional process mining approach described above is to train a neural network model to learn the behaviour of the event sequences, then use the trained model to make predictions [6]. Neural networks are trained using a set of data as follows. When a neural network outputs a value in response to some input, this predicted output value is then compared with the actual output value in the data. A loss function is defined as the function for the difference between the predicted and actual values. An algorithm called back-propagation is used to minimize this loss value.

A Dense layer in a neural network is a layer where all the nodes are connected to all the nodes in the previous layer. Normally, the output layer of a neural network will be a Dense layer. In the case of a regression problem, with numerical output, each node in the output layer outputs a numeric value. For the case of classification, each node will correspond to a different class, and the softmax activation function is used to convert every numerical output to a probability of that class occurring. In effect, the output layer outputs a probability distribution vector over the number of different classes. The loss function used for (non-binary) classification is known as the categorical-crossentropy loss function.

D. Recurrent Neural Networks (RNN)

When processing sequences using a neural network, the sequence is fed into the network over a number of time-steps. The network is required to learn sequential behaviour, how events at one time-step affect the subsequent events in the sequence. An RNN is a neural network where the output from the hidden layer is fed back into the hidden layer on the subsequent time step as shown in Figure 2. In this way, the occurrence of particular events in a sequence can affect the likelihood of other events occurring later in the sequence.

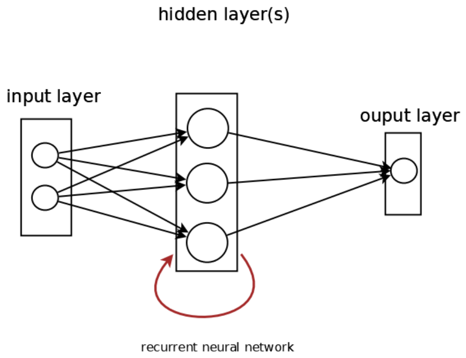


Figure 2. RNN with a single hidden layer.

In order to train the RNN, the required input is a multi-dimensional array of the shape (sequences, time-steps, features). The length of the first dimension is the number of sequences, or cases, to train. The length of the second dimension is the number of time steps within a case of chronological ordered events. When the input is a categorical variable, (an event type in this project), this categorical variable is one-hot encoded and the length of the third input dimension is equal to the total number of event types.

The shape of an RNN network is normally the size of a probability distribution over unique event types for every sequence. Note that it is possible to configure a network to output a prediction at each time step of a sequence. In that case, the output is of the shape (sequences, time-steps, features) and is essentially a sequence of predictions/probability distributions, one for every time step in every sequence. In Keras, this is achieved by including a TimeDistributed wrapper layer around the Dense layer that produces a prediction at every time slice. This is used in the teacher forcing method outlined in Section IV below.

E. Long Short Term Memory (LSTM)

A LSTM model is a RNN model that supports long-term dependencies in noisy, sequential data [15]. LSTM nodes are no longer simple single nodes but rather a sub-network of other nodes and activation functions. Long term dependencies are captured using three **gates** in an LSTM node, an input gate, a forget gate and an output gate. These gates control how data from a previous time step is used, stored or thrown away. The forget gate determines which data is to be discarded. The output gate determines the output based on previous input and the state of the LSTM node. Event sequences in process mining data can contain long term dependencies and hence LSTM networks are useful for modeling such data.

IV. Model Training Methods

We examine two approaches taken when training an LSTM model to predict future events within a case, the prefix method and the teacher forcing method.

A. The Prefix Method

This approach [5] generates a set of all possible prefixes longer than the length of a single event from all sequences to train the model. For example, for the sequence of event types 1,2,3,4,5 the input and output is shown in Table I.

TABLE I. Input Prefixes and Output.

X (input)	*y (target)*
[1, 2]	3
[1, 2, 3]	4
[1, 2, 3, 4]	5
[1, 2, 3, 4, 5]	!

Effectively, the network is trained to predict a target value *y* given the input value *X*. Note that the model is trained to predict a *!* character, which denotes the completion of a case. The trained model is then used to predict a single suffix event following the prefix. Following the work of Tax et al. [5] the shortest prefix used is of length two, so predictions start after the second event.

The architecture for the model is shown in Figure 3.

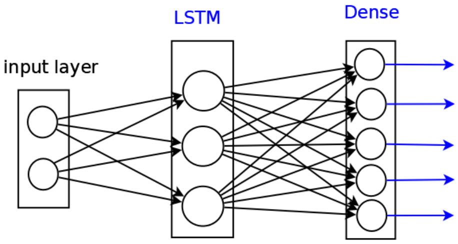


Figure 3. LSTM network with a Dense output layer.

ISBN: 978-1-61208-785-6

B. Teacher Forcing Method

Teacher forcing uses the ground truth from a prior time step in a sequence as input [7][16]. For the sequence 1,2,3,4,5, the input and output for the neural network are shown in Table II.

TABLE II. INPUT AND OUTPUT FOR TEACHER FORCING.

X (input)	*y (target)*
[1, 2, 3, 4, 5]	[2, 3, 4, 5, !]

Notice that input sequence *X* is the event sequence with the last event removed and output *y* is the sequence with the first removed. For training, *X* is passed through the LSTM network once. The input at each of these time steps is the ground truth value at the previous time step. For example, while predicting 2 the input is 1, and so on.

The architecture of the model is as shown in Figure 3 except that the output Dense layer has a TimeDistributed wrapper layer. The existence of a TimeDistributed layer in the network distributes the output Dense layer with the softmax activation function to every time step in a sequence, resulting in a prediction at every time step. Note that training can occur on one pass of the sequence through the network, as opposed to a pass for every prefix of a sequence as with the prefix method.

V. EXPERIMENT AND EVALUATION

In this section we introduce the data sets used. We then outline the model training and evaluation approaches for both the prefix method and the teacher forcing method.

A. Data Sets

The above techniques are applied to four data sets used in the project, the BPI challenges from 2012, 2013, 2014, and the Helpdesk data set used by [5].

1) BPI 2012 Data Set: The BPI 2012 data set comes from the BPI challenge workshop in 2012 [8]. The data set is an event log taken from an application procedure for financial services, such as a personal loan or overdraft, at a large Dutch financial institution. Originally comprised of several sub-processes, the event log is narrowed down to the 'work item' sub-process and only cases with the work item events types *start* and *complete* were included. This reduction of the sub-processes has been previously used by [5] and [17].

The resulting event log contains a vocabulary of 6 event types and 7469 cases. Each event type is defined by an *Activity ID* in the data. Each case, or process sequence of *Activity ID*s are identified and grouped by their *Case ID*, which acts as a unique case identifier.

2) Helpdesk Data Set: The Helpdesk data set is an event log from a ticket management process for an Italian software company's help desk. Tax et al., [5] used this data set as supplementary material. The log consists of 9 different event types, 3804 cases and 13710 events. The different event types are represented by their *Activity ID* with the *Case ID* being the unique case identifier. The list of cases are returned by grouping all *ActivityID*s by their respective *CaseID*s.

3) BPI 2013 Data Set: The BPI data set for 2013, provided by the BPI 2013 workshop [9], is an event log for an incident management system called VINST. VINST solves IT related problems for Volvo Information Technology. Each problem or IT service request made to VINST is treated as a case with the Service Request number being the case identifier. The resulting event log was provided by Volvo IT Belgium and lists 7553 cases and the designated case identifier is the column labeled *SR Number*. For both this research project and [18], the vocabulary of event types are defined by generating every unique possible combination of the two columns *Status* and *Sub Status*, returning a vocabulary of 13 event types. Every event type is then mapped to an event number. Each different trace of event numbers is then grouped by the *SR Number* to return the data set.

4) BPI 2014 Data Set: The BPI 2014 data set is an event log selected from a collection of three different processes investigated by the ICT department for Rabobank Group, a banking and financial services company. A service management tool logs customer support calls for software support. The service management tool logs three main sub-processes, which are outlined below and are provided in CSV by Rabobank.

- *Interactions*: calls are made by customers (Rabobank colleagues) to the Service Desk where a Service Desk Agent (SDA) answers, resolves the issue for the customer and logs these calls as an *Interaction* or assigns the technical issue to an Assignment Group
- *Incidents*: the SDA is unable to resolve a customer call, and based on a given urgency and impact, assigns the issue to an Assignment Group to solve and the process to solve the issue is logged by Rabobank as an *Incident*; each incident is treated as a case of logged activities an assignment group takes to resolve said disruption
- *Changes*: if a service disruption were to occur more than once then a problem analysis investigation is launched that will lead to an improvement plan to prevent the service disruption from happening again subsequently logging a *Change* record

The primary sub-process investigated by Rabobank selected for this research is the *Incident* data. The actual data set used is a translated event log built from csv files relating to every incident. The files downloaded from the BPI 2014 workshop for this project are: *Detail Incident.csv*, a list of 46607 unique incidents, and the *Detail Incident Activity.csv* file, an activity log of recorded events related to 46605 incidents in the list of incidents. For each individual incident, the column *IncidentID* is the designated case identifier. To define the vocabulary of unique events for the data set, the columns *Category* and *IncidentActivity-Type* are selected from the list of cases and incident activity log respectively. The two files are then merged into a new singular event log using the *IncidentID* column as a joining key. The two aforementioned columns *Category* and *IncidentActivity-Type* are now both in the same event log. The vocabulary of different event types can now be now defined using every possible unique combination of the two columns.

This returns a vocabulary of 91 event types in total. Each trace of events is then grouped by the column *IncidentID* to return the data set of incidents, or cases.

B. Model Training and Evaluation for the Prefix Method

All the training prefixes are generated from the data set as outlined in Section IV above. The prefixes are pre-padded to the length of the longest case. Training parameters include batch size and the number of epochs. Finally, 20% of the training data is set aside for validation purposes allowing us to see the behaviour of the training/validation and accuracy/loss values at the end of each epoch.

When evaluating the trained model, prefixes are generated for every case in the test data set. Each prefix sequence generated is passed through the trained network and the model outputs the probability distribution vector over the number of different event types. To determine the predicted event, the index of the largest value in the probability vector is found and its respective event type is returned. The accuracy of the model's performance is found by comparing the predicted event type with the actual event type for every testing prefix.

C. Model Training and Evaluation for the Teacher Forcing Method

The training sequences are sorted in increasing size and the data is divided into mini-batches of a chosen size. The model training method is then called on each of these mini-batches. All sequences in a batch must be of the same length, so all sequences are pre-padded with zeros to the size of the longest sequence in the mini-batch. Mini-batches will typically have different lengths.

To evaluate the model using the Teacher Forcing method, each sequence in the testing data is fed through the network. The output is a sequence of probability distributions corresponding to a prediction for every time step. Accuracy is evaluated by comparing predicted events with the subsequent event in the input sequence starting after the second event.

VI. Results

Having built an LSTM based model, a range of parameter values were evaluated to find the optimum configuration of meta-parameters for the model. Table III gives model configurations and prediction accuracy for the BPI 2012 data using the Prefix version. (Other tables for the other data sets and Teacher Forcing method are not included in the paper.) Notice that the use of a second LSTM layer or adding a Dropout layer for regularization did not improve the accuracy.

These models used an Adam optimizer [19], which is efficient and requires minimal memory and parameter tuning, and works well with cases comprised of noisy data. The optimizer also uses an adaptive learning rate, a hyper-parameter that controls the step size at each iteration of the training algorithm. It is a trade off between reaching an optimal solution in a timely manner, and overshooting the optimal solution.

The maximum accuracy values for each data set are listed in Table IV.

TABLE III. Prefix Method on BPI 2012 Data Set.

LSTM	Dropout	Nodes	Batch Size	Epochs	Accuracy
1	0	100	10	50	65.68%
1	1	100	10	50	66.31%
2	0	100	10	20	66.34%
1	0	100	6	20	66.60%
1	0	120	32	20	67.73%
1	0	60	32	20	67.88%
1	**0**	**100**	**32**	**20**	**68.64%**

TABLE IV. Maximum Accuracy for the Data Sets.

Data Set	Cases	Events	**Max Acc.**
BPI 2012	7469	6	68.64%
Helpdesk	3803	9	81.16%
BPI 2013	7553	13	65.66%
BPI 2014	6000	69	48.28%

As expected, it is harder to make predictions for data sets with a larger number of event types. The Helpdesk data set seems to be the exception with a vocabulary of 9 different event types and the highest accuracy of 81%. We looked at the frequency distribution for event types, including the end of case character *!* in this data set. A graph of the frequency distribution is shown below in Figure 4.

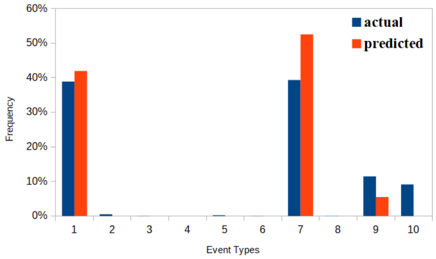


Figure 4. Frequency Distribution of Helpdesk Events.

Six event types are quite infrequent compared to the others and we could say the effective number of event types is four, including the end of case event, and this explains the high accuracy. The Teacher Forcing method makes no predictions for the other infrequent event types, while the Prefix method makes 5 predictions of these infrequent events out of a total of 1267 predictions. Notice that both models over predict the more frequent events while under predicting those rarely appearing. This is to be expected as the neural network has not seen enough of these infrequent events to learn how to predict them.

A similar situation holds for the 2013 data set where there are five infrequent event types. Figure 5 shows the frequency distributions for the actual event types in the data set and the predicted events given by the two methods. Notice that again frequent events tend to be over predicted and infrequent events tend to be under predicted. Note that the Prefix version is also less prone to this bias than the Teacher Forcing version.

Table V displays the accuracy and execution time for each of the four different data sets and two training methods.

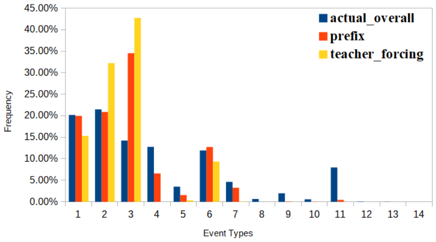


Figure 5. BPI 2013 Frequency Distribution of Test Events.

TABLE V. PREFIX AND TEACHER FORCING COMPARISON.

Data Set	Prefix Method		Teacher Forcing Method	
	Acc.	Time (mins)	Acc.	Time (mins)
BPI 2012	68.64%	17.9	68.18%	0.5
Helpdesk	81.16%	2.7	80.39%	0.97
BPI 2013	65.66%	32.1	62.94%	0.5
BPI 2014	48.28%	42.5	43.68%	0.5

The teacher forcing method is much faster to train. As shown in Table I for a case of length n, n-2 prefixes are generated for the prefix method. So, roughly speaking the number of training instances for the prefix method is an order of n times larger and this accounts for the substantial difference in training times. As shown, both techniques generally produce similar results for the 2012 and Helpdesk data sets. The prefix method produces better results for the 2013 and 2014 data sets. This is surprising as the loss function is the same. Further work is required to understand why this is so.

The full BPI 2014 data set contains 46606 cases. Training using the full data set was carried out on an NVIDIA GPU server with a four-card Tesla SXM2. For the Prefix method, using the full data set resulted in a 49.49% accuracy and took an hour and 7 minutes to train.

VII. CONCLUSION AND FUTURE WORK

CSIRTs constantly struggle to attend to the large number of alerts generated by intrusion detection software. One approach to this is to identify critical services and assets in the organisations that are being targeted. If suitable models exist linking business processes and supporting infrastructure, the ability to predict the next case activities can support CSIRTs in prioritizing the examination of intrusion alerts.

This paper evaluates the use of LSTM neural networks for predicting next activities in a case. In particular it looked at four data sets and two training methods. Prediction accuracy for the different data sets depends on, to a large extent, the number of event types in each data set. As expected, it is harder to predict event types where there is a large number of them. Also models tend to under predict rare events and over predict common events. This bias was more pronounced when the model was trained using the teacher forcing method.

The prediction accuracy for the Helpdesk data set was the highest at 81.2% for nine event types. This is high compared to the BPI 2012 data set, which only had six event types and an accuracy of 68.6%. However, five event types from the Helpdesk data set had very low frequency resulting in the model mostly choosing between four different event types. This explains the higher accuracy obtained.

Comparing the two methods of training we saw that in two cases the accuracy was nearly the same (within 1%). In the other two the prefix method was slightly higher, 2.8% better for the BPI 2013 data set and 4.6% for the BPI 2014 data set. Even though the differences were small this was slightly surprising as we expected the results to be the same. In all cases, the teacher forcing method takes an appreciably shorter time to train, by a factor of up to six times faster.

To the best of our knowledge, LSTM networks have not been previously applied to event prediction for the 2014 data set. For the Helpdesk data set, our accuracy results were 10% better than published by Tax et al. [5]. For the BPI 2012 data set our results were 8% lower. It should be stressed that we are not comparing our means of using the timestamp and event types as input for the LSTM model to Tax's method.

REFERENCES

[1] S. C. De Alvarenga, S. Barbon Jr, R. S. Miani, M. Cukier, and B. B. Zarpelão, "Process mining and hierarchical clustering to help intrusion alert visualization," *Computers & Security*, vol. 73, pp. 474–491, Mar. 2018.

[2] S. Mauser and T. Eggendorfer, "Detecting security attacks by process mining," *Algorithms and Tools for Petri Nets*, pp. 1–33, Oct. 2017.

[3] S. C. de Alvarenga, B. B. Zarpelão, S. Barbon Jr, R. S. Miani, and M. Cukier, "Discovering attack strategies using process mining," *AICT*, vol. 15, pp. 119–125, 2015, ISBN: 978-1-61208-411-4.

[4] W. M. Van der Aalst and A. K. A. de Medeiros, "Process mining and security: Detecting anomalous process executions and checking process conformance," *Electronic Notes in Theoretical Computer Science*, vol. 121, pp. 3–21, Feb. 2005.

[5] N. Tax, I. Verenich, M. La Rosa, and M. Dumas, "Predictive business process monitoring with lstm neural networks," in *International Conference on Advanced Information Systems Engineering*. Springer, Jun. 2017, pp. 477–492.

[6] J. Evermann, J.-R. Rehse, and P. Fettke, "Predicting process behaviour using deep learning," *Decision Support Systems*, vol. 100, pp. 129–140, Aug. 2017.

[7] K. Drossos, S. Gharib, P. Magron, and T. Virtanen, "Language modelling for sound event detection with teacher forcing and scheduled sampling," *arXiv preprint arXiv:1907.08506*, Jul. 2019.

[8] B. van Dongen, "Event log of a loan application process," Eindhoven University of Technology, 2012, URL: https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f/ [accessed: 2012-04-23].

[9] W. Steeman, "Bpi challenge 2013, incidents (2013)," Ghent University, 2013, URL: https://doi.org/10.4121/uuid:a7ce5c55-03a7-4583-b855-98b86e1a2b07/ [accessed: 2013-04-12].

[10] B. Van Dongen, "Bpi challenge 2014 (2014)," Rabobank Nederland, 2014, URL: https://doi.org/10.4121/uuid:c3e5d162-0cfd-4bb0-bd82-af5268819c35/ [accessed: 2014].

[11] R. J. C. Bose, E. H. Verbeek, and W. M. van der Aalst, "Discovering hierarchical process models using prom," in *International Conference on Advanced Information Systems Engineering*. Springer, Jun. 2011, pp. 33–48.

[12] F. Silva and P. Jacob, "Mission-centric risk assessment to improve cyber situational awareness," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, Aug. 2018, pp. 1–8.

[13] W. M. van der Aalst, C. Günther, J. C. Recker, and M. Reichert, "Using process mining to analyze and improve process flexibility-position paper," in *The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium*. Namur University Press, 2006, pp. 168–177.

[14] W. Van der Aalst, T. Weijters, and L. Maruster, “Workflow mining: Discovering process models from event logs,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128–1142, Jul. 2004.

[15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[16] A. M. Lamb *et al.*, “Professor forcing: A new algorithm for training recurrent networks,” in *Advances In Neural Information Processing Systems*, 2016, pp. 4601–4609.

[17] R. J. C. Bose and W. M. van der Aalst, “Process mining applied to the bpi challenge 2012: divide and conquer while discerning resources,” in *International Conference on Business Process Management*. Springer, Sep. 2012, pp. 221–222.

[18] N. Mehdiyev, J. Evermann, and P. Fettke, “A multi-stage deep learning approach for business process event prediction,” in *2017 IEEE 19th Conference on Business Informatics (CBI)*, vol. 1. IEEE, Jul. 2017, pp. 119–128.

[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, Dec. 2014.