Research Projects

ERIC CHEN, University of California, Los Angeles

In this document, I include most of the research that I have done for the past three years. Because of the heterogeneity nature of my research, I list them chronically. The past research makes me diverse and versatile, but I am always open to learning new things and "unlocking" new skills.

For every project, I describe the following aspects: (a) the **motivation** of the project (b) **Overall Design** that captures the essence of each work (c) **My contribution** to this project (d) **Publication URL** reference (e) **Implementation** environment and testbed (f) If applicable, whether and how this project is **deployed in real world**.

Contents

1 WIRELESS NETWORK

All my research on wireless network is done upon instruction and supervision of Professor Songwu Lu in his Wireless Networking Group(WING). One of my major works is Resilient Multicast, which improves the application layer throughput of Wi-Fi multicast over 450 times and complies with current 802.11 standard. Also, we hacked Android 9's Wi-Fi driver to open up the blackbox of current Wi-Fi design, so that we can perform learning and reasoning on user's current cause for Wi-Fi failures. I also implemented LTE cross layer analysis and analyze the packet dependency among LTE's different layers and performed analysis on them.

1.1 Resilient High Rate Wi-Fi Multicast

Motivation. Wi-Fi is a shared broadcast channel ideal for multi-user gaming and video streaming. However, its multicast basic rate is limited to a 1 Mbps by legacy IEEE 802.11 standards; improvements are scant purely hacking on this rate. Resilient Multicast, which improves the application layer throughput of Wi-Fi multicast over 450 times and complies with current 802.11 standard. We identify five issues related to current Wi-Fi multicast design.

(1) Wireless multicast runs at the lowest basic rates
(2) 802.11 power-save mode on multicast
(3) No A-MPDU frame aggregation for Multicast frames
(4) Higher channel width is not enabled for multicast frames
(5) No retransmission mechanism for multicast

Overall Design. Our design focuses on enabling the Layer 2 designs that are not currently available to Wi-Fi and a retransmission mechanism with its corresponding Layer 2.5 headers.

(1) Enabling high MCS rates. The current Wi-Fi multicast runs at the lowest supported data rate (1Mbps for 802.11g and 6Mbps for 802.11a). The most recent 802.11n/ac standards use MIMO and higher MCS (modulation and coding scheme) rates.
(2) Enabling A-MPDU frame aggregation for multicast One of the pivotal enabling features in 802.11n/ac for high speed transmission is the frame aggregation. By enabling frame aggregation for the multicast frames, resilient multicast frames are aggregated into super-frames for transmissions.
(3) Expanding multicast operation mode for 40 MHz channel widthIn the current 802 wireless media, if the channel width is doubled (for example, from 20MHz to 40MHz), the maximum physical-layer (PHY) rate should double with the same MCS rates.Resilient multicast enables the higher channel operation for the multicast frames.
(4) Enabling Retransmission Mechanism The current Wi-Fi multicast does not use any retransmission mechanism. To increase the robustness of Wi-Fi multicast, we have designed resilient multicast mechanisms based on the current 802.11n standard. To this end, we have added a custom L2.5 header, which sits between Layer 2 and Layer 3r. We consequently manage the multicast frame retransmissions using this new header. The header structure is shown in Figure 1 and the procedure is shown in Figure 2.

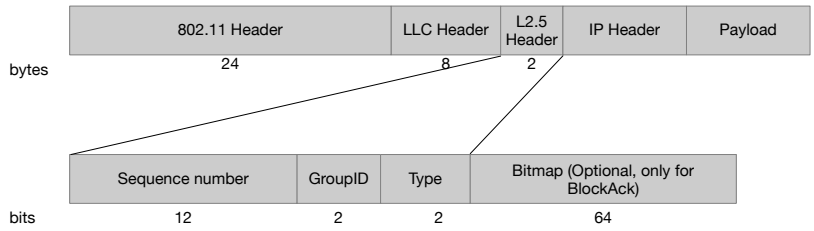

Fig. 1. L2.5 Header Structure

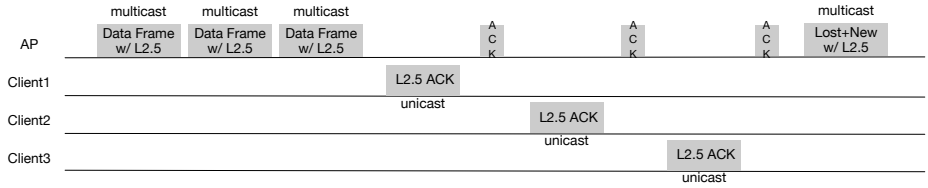

Fig. 2. Procedure of Resilient Wi-Fi Mulitcast

Upon receiving aggregated multicast frames sent from the AP, each client tracks the Sequence Number in the L2.5 Header, and records whether each individual frame carried by this aggregated frame has been successfully received or not. After receiving multiple aggregated multicast frames (64 frames in current design), the clients unicast the BlockAck message back to the AP, which include the bitmap to indicate the status of each individual frame received during this interval. Since the BlockAck is a unicast frame, it will receive the acknowledgment from the AP. The current Wi-Fi standard provides the retransmission mechanism for the unicast frames.

My Contribution. I started to work on this project since November 2018. In this project, Professor Songwu Lu offers instructions on conceptual level, PhD Candidate Zengwen Yuan offers instructions on implementation and I worked with Phd Candidate Jinghao Zhao on design, implementation and write-ups. We are currently preparing an IETF RFC draft for this project.

URL. The design and implementation details is currently uploaded to UCLA Computer Science Report. http://fmdb.cs.ucla.edu/Treports/Multicast_Technical_Report.pdf

Implementation. This project is implemented based on mac.80211 driver and Ath9K driver on openWRT. The experiments are carried on TP-Link N750 as AP, and heterogeneous devices such as Google Pixel, XiaoMi MIX2 and Microsoft Surface tablet as STA. IPerf2 multicasts UDP data packets to IPv4 multicast addresses, which clients are all connected to this address. Power-save mode is disabled among all clients.

1.2 Intelligent Wi-Fi

Motivation. We have identified three issues regarding the intelligence of the current Wi-Fi technology: poor learnability, weak information exchange, and poor reasoning capability.

The current Wi-Fi largely operates as a blackbox for higher-layer designs, in both the current AP management and smartphones. Low-level MAC & PHY information cannot be readily acquired and analyzed by

phone users, and users thus do not know how and why failures arise or mobility is triggered. In addition, it is hard for both the AP managers and phone users to know where things go wrong. For example, low throughput can caused by either user mobility or by AP misconfigurations. However, in the current Wi-Fi operations, these conditions cannot be acquired and informed to each other.

Since the current Wi-Fi does not enable exchanges on important information, AP and phone users are virtually blind to each other. AP can acquire some basic user information such as device RSSI, but it lacks valuable information such as mobility and signal strength relative to other APs. Such information is critical to AR/VR applications such as interactive AR and real-time streaming applications, which require high throughput and stable channel conditions. Even if the current AP design may collect some device-side information such as device RSSI, it cannot perform some basic reasoning tasks given the collected information, and make intelligent decisions (for example, to perform multicast roaming between different APs).

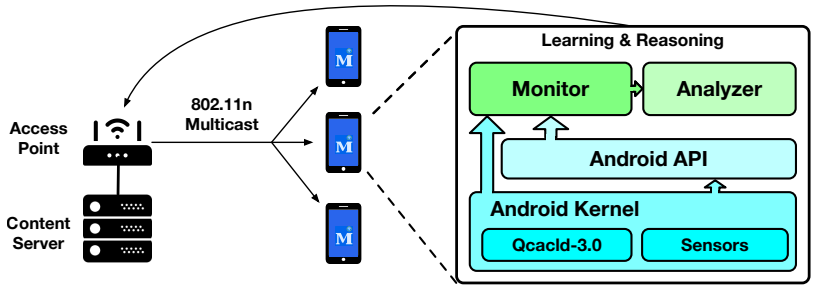

Fig. 3. A basic design of Intelligent Wi-Fi

Overall Design. In our design, we open up the network information access to the blackbox Wi-Fi, and conduct data analytics to enhance critical services such as multicast roaming between different APs. We first design our learning module by exposing raw logs from different sources to the device user-space. For our essential Wi-Fi logs, it requires modifying kernel Wi-Fi drivers. The learning module parses the raw logs and extracts the carried information. It also learns other side information from system APIs, such as mobility, and remaining battery level, to facilitate learning.

Given the learned information from heterogeneous sources, our reasoning module is designed to reveal the operational dynamics behind the Wi-Fi multicast. Based on the learned information, we perform reasoning by exploiting previously learned results and domain knowledge. The reasoning module infers various conditions, such as weak channel condition, and mobility. The inferred results are sent as feedback messages to the AP to further facilitate the AP make intelligent decision.

My Contribution. I started to work on this project since March 2019. This project is associated with aforementioned Resilient Multicast, but gives a more intuitive interface for phone users to observe current Wi-Fi's status based on our learning and reasoning results. Professor Songwu Lu offers instructions on conceptual level, PhD Candidate Zengwen Yuan offers instructions on implementation and I worked with Phd Candidate Jinghao Zhao on design, implementation and write-ups.

URL. The design and implementation details is currently uploaded to UCLA Computer Science Report. http://fmdb.cs.ucla.edu/Treports/Multicast_Technical_Report.pdf. The poster can be found on http://keplerc.com/publications/Multicast/multicast-poster-v5.pdf.

Implementation. We have implemented the above design on Android Phones as the clients and TP-Link N750 router as the wireless AP. At the phone side, we modify Qcacld 3.0 driver to print out more information and replace the original Pixel 2 driver by Magisk. We implement our Android learning module to collect driver logs and Android system APIs. The application also collects other side information such as battery level and the status of mobility (static, walking, etc.), and analyzes the collected information by performing reasoning tasks. Once reasoning at the phone is completed, it sends out feedback messages to the AP. In our current prototype, we consider the influence of mobility of smartphone users, and the channel strength to the multicast frame loss rate. A sample screenshot can be found in Figure 4.

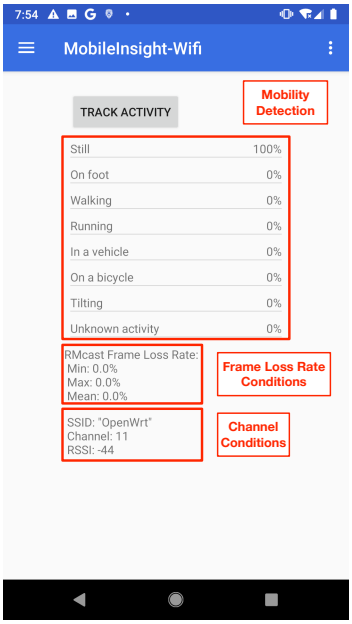

Fig. 4. A Screenshot of Our Prototype Phone Application

1.3 LTE Cross Layer Analysis

Motivation. A unique issue is how to handle the protocol dependency in the sampling. Figure 6 exemplifies one scenario at the data plane. Each data packet should traverse across link-layer and physical-layer protocols, and can be possibly divided into multiple segments for wireless (re)transmission. This results in the dependency between link-layer packets and physical-layer segments. Setting one layer's sampling rate may impact the other layers. To this end, we seek a simpler, automatic approach: We track the cross-layer message dependency, set the highest layer's sampling rate only and tracking the dependency, and automatically set the lower-layer's sampling rate based on the dependency model.

My Contribution. In this project, I implemented LTE cross layer analysis and analyze the packet dependency among LTE's different layers and performed analysis on them. This project is worked with PhD candidate

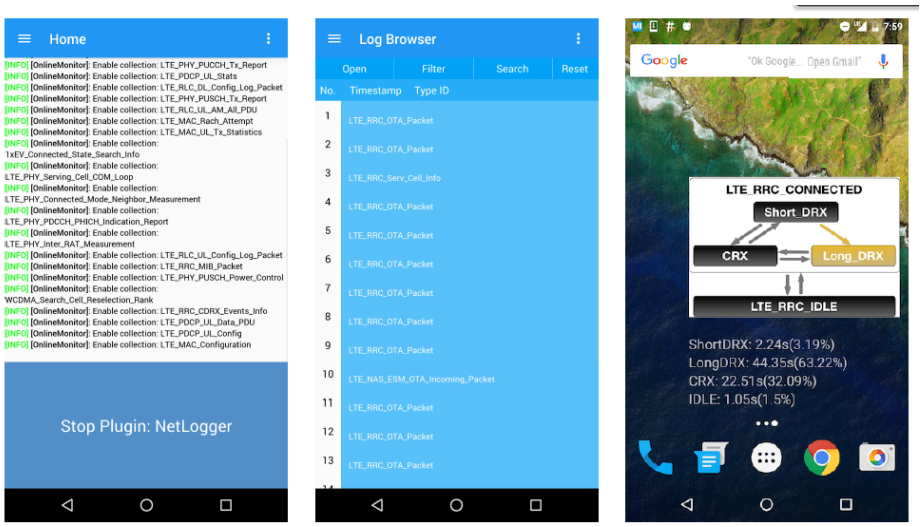

Fig. 5. Overview of MobileInsight, our in-phone mobile network analytics tool

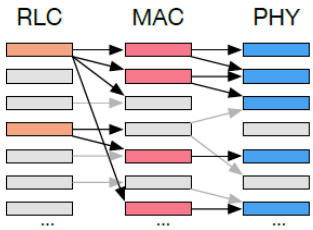

Fig. 6. Coordinated sampling for energy-efficient analytics

Zhehui Zhang. In this project, she walked me through the procedures on how different layers of LTE(PDCP, RLC, MAC) coordiante with each other. I developed efficient sampling algorithms for the cross-layer analytics.

URL. The open source code is on http://mobileinsight.net/. It contains all the code that we built and keep maintaining. The design itself is an unpublished branch.

Implementation. The platform for analysis is MobileInsight(mobileinsight.net). A sample figure can be found in Figure 5. In order to let every device monitor and analyze âĂIJblack-boxâĂİ mobile network operations. This calls for runtime, fine-grained information (protocol states, parameters, and operation logics) from full-stack network protocols (physical/link layer, radio resource control, mobility management, data session management) inside the commodity phones. Unfortunately, no existing approach can meet this requirement. Mobileinsight is a perfect platform for this in-phone mobile network analytics task.

The parsing script is written in Python, and the rough length of the whole analyzer is approximately 2000 lines.

2 INFORMATION NETWORK

2.1 GloGCN

Motivation. Graph Convolutional Network (GCN) has demonstrated state-of-the-art performance on semi-supervised node classification. By adapting the convolution filter concept from Convolution Neural Network (CNN), it effectively leverages rich semantics encapsulated between nodes and edges in a graph and has gained great attention in even practical fields such as physical systems, chemistry, and text mining.

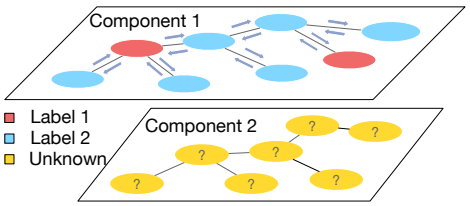

Fig. 7. An illustration on the limitation of GCN when propagating across components.

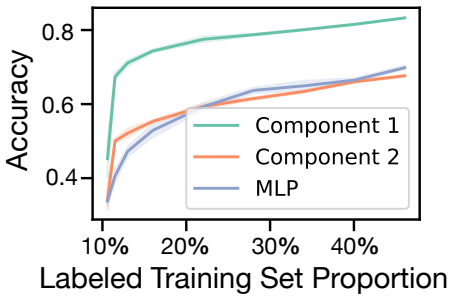

Fig. 8. We duplicate the Citeseer dataset in a single graph as Component 1 (partially labeled) and Component 2 (not labeled) corresponding to Figure 2(a). Even when the labeled training set takes a large proportion of Component 1, the accuracy on Component 2 is still similar to that of MLP.

However, GCN faces a serious drop in accuracy when trained on graphs having a low proportion of labeled nodes, which strongly limits GCN for practical usage. The *locality limitation* of GCN accounts for this downfall: its graph convolution layer can only propagate label information to nearby nodes, so when labeled training examples are sparse on a graph, nodes that are distant from labeled training cannot be propagated effectively. The situation is even worse when the graph is not connected. For example, the nodes on a component with no labeled examples usually demonstrate a similar accuracy to that of Multi-Layer Perceptron (MLP), a model that ignores the connectivity information and classifies nodes solely by node attributes. Intuitive fixes such as stacking deep layers are ineffective or even harmful due to the nature of GCN as Laplacian smoothing. As a result, enabling an effective global propagation is the key to the accuracy of GCN in this setting.

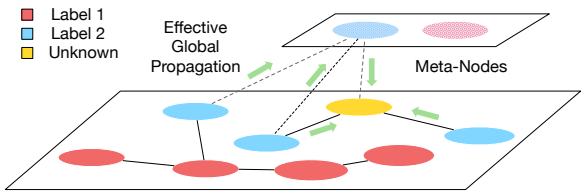

Fig. 9. The basic mechanism of GloGCN on label propagation. Labels can be propagated globally from both nearby nodes and distant similar nodes through meta-nodes.

Overall Design. The design of GloGCN is shown in Figure 9. We construct a group of self-learned *meta-nodes* so that every meta-node has strong edges only to nodes that are similar to it. GloGCN uses these edges to bridge similar but distant nodes in a graph and facilitate label information to propagate globally with minimal distortion to the original graph. In addition, in order to model and optimize this edge weight distribution and update meta-nodes, we jointly train the inference structure of GCN with an unsupervised loss based on a community-aware target edge weight distribution.

Our model significantly improves the accuracy on various datasets. For example, our model improves GCN up to 10.3% on Cora and 10.4% on Citeseer datasets when the number of labeled training nodes is less than 100. We also observe our model maintains a much clearer embedding space even when labeled training nodes are limited.

My Contribution. This work is a class project of graduate level data mining course. I designed, implemented and conducted all the experiment. Ph.D. candidate Zeyu Li(who was my TA) helped me with writing, Professor Yizhou and Professor Wei Wang provided valuable instructions.

URL. The report is on http://keplerc.com/publications/AAAI2020/AAAI_protected.pdf. PASSWORD for this pdf is **apply**. Note that this report is still not published so please keep it confidential.

Implementation. The code is implemented on Python and compared with original GCN. Cora dataset is one of the most popular citation network datasets. Compared with various state-of-the-art models in Table 1, our models has the highest accuracy especially when number of labeled nodes is small.

Table 1. Classification Result on Cora Dataset

# examples	20	35	50	80	100	200	300	400
	0.7%	1.2%	1.8%	2.9%	3.6%	7.3%	11.0%	14.7%
Random	14.2	14.2	14.2	14.2	14.2	14.2	14.2	14.2
MLP	24.7	40.7	45.6	52.9	59.6	61.3	62.4	63.2
LP	42.9	45.5	65.5	66.5	68.1	76.3	78.1	76.5
GraphSage	50.4	61.5	69.4	76.2	76.4	79.9	79.2	81.1
GPNN	56.9	64.5	73.8	78.1	77.9	84.0	84.5	84.1
LCN	51.2	61.0	69.2	71.5	72.3	80.7	81.3	81.7
DGCN	57.1	72.4	76.2	78.7	78.2	81.8	83.1	82.9
Cheby	44.0	56.3	65.2	73.6	78.4	81.9	82.7	83.2
GCN	54.2	64.3	74.2	77.1	78.1	84.5	85.0	84.8
GCN-V	53.9	64.6	74.2	77.5	77.9	84.0	84.8	84.5
GloGCN	58.5	73.8	79.9	81.1	81.3	84.9	**85.5**	**85.6**
GloGCN-V	**60.0**	**74.9**	**80.4**	**81.5**	**81.8**	**85.4**	85.1	85.6

3 DEEP LEARNING SYSTEMS

3.1 Out-of-Distribution Detection

Motivation. We consider the following scenario: Alice is taking an exam. She encounters a multiple choice question that she has never met in textbooks and thus she has low expectation on answering it correctly. To avoid an extra cost for answering it wrong, she decides to skip it and continues working on those that she practiced a lot. Machine learning models should do the same. While machine learning algorithms have gained great success in making predictions on data points frequently appear in training set, they are troubled by testing data points that rarely occur in training data. If we mistakenly feed a wrong query to any state-of-art machine learning models, they will assign the data point to an undefined label, but the logical answer should be simply "I don't know".

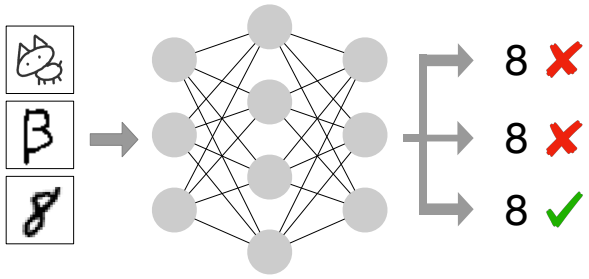

Fig. 10. A traditional MNIST prediction model that pretends it knows the answer, but what it should return is simply "I don't know"

Due to limitations on available data collection methods, what neural networks learn should never be expected to deal with all the scenarios: predicting on samples that rarely appear in training set will have very low accuracy.

Overall Design. we design an end-to-end neural network. It learns an inherent discriminative embedding on the training set to perform out-of-distribution(OOD) detection and classification at the same time: both OOD data points and points that resemble those with different labels can be visually observed in this embedding space. Based on this model, we also devise a training scheme that trains on only inliers. Experiments on various datasets and metrics validate that our method outperforms the state-of-art OOD detector. We design our architecture in Figure 13 based on three losses: reconstruction loss, classifier loss and also an embedding penalization. For complete and rigorous problem formulations and derivations, please refer to my publication.

My Contribution. This work is a class project of graduate level machine learning course. THIS IS MY SOLO WORK except valuable discussions from Professor Quanquan Gu. My labmate Jinghao Zhao helped polish this work.

URL. The paper is published in IJCNN 2019. Paper url is on http://keplerc.com/publications/IJCNN2019/IJCNN_Paper.pdf and the poster is on http://keplerc.com/publications/IJCNN2019/IJCNN_Poster.pdf

Implementation. We implemented it on Python and Keras. The datasets that I worked with are MNIST, CIFAR 10, iSIN, Omniglot, notMNIST, CIFAR10-bw and I also defined my own metric, which all other state-of-the-art models fail: MNIST-{x} This dataset is generated from original MNIST dataset by removing points

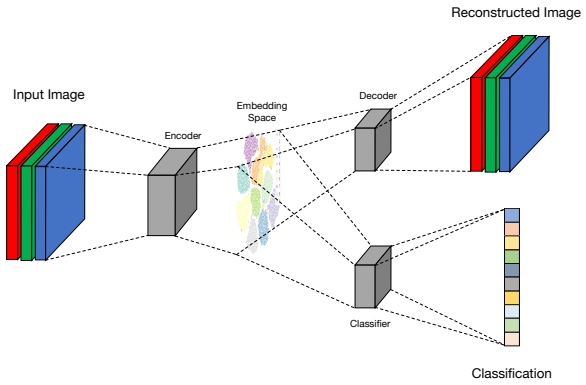

Fig. 11. Architecture of my design

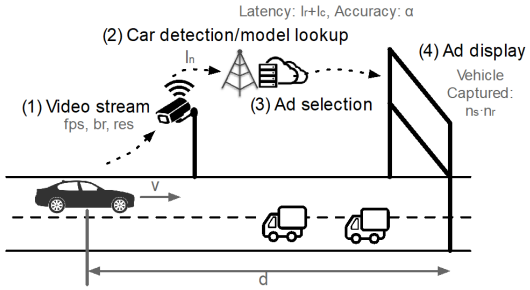

Fig. 12. The problem formulation for LTE target advertisement that we consider

with label x. For example, MNIST-{0,1} means we use data points with label 2,3,4,5,6,7,8,9 as in-samples and points with label 0 and 1 as outliers.

3.2 LTE Target Advertising

Context and Formulation. The basic formulation in shown in Figure 12. Our system needs to offer the digital advertisers real-time, targeted ads for highway drivers based on car models. It deploys the roadside cameras ahead of the digital billboard to capture the passing cars' images. Our patent supports two deployment modes. In the first one, the camera streams the video with the cars the edge. In the second mode, the local camera is able to capture the frame and compress it as an image. The image is sent to the cloud via socket. The edge will detect the vehicles make and model in the frames and classify the car model. It then selects the appropriate advertising to put on the billboard as that car passes. As the car passes, the billboard's ads rotate. Faster targeted ads can rotate more often, allowing billboard advertiser to sell more ads.

Overall Design. We designed an adaptive system and method for optimizing the wireless-powered outdoor targeted advertising may include an outdoor digital billboard, an IoT camera with video or image quality adaptor and network information manager, and an advertisement server with video or image configuration profiler, offline/online profiler, model detection/recognition engine, and advertisement controller. The IoT

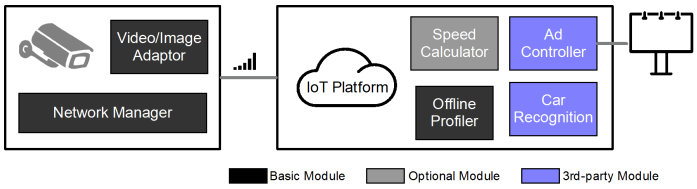

Fig. 13. Architecture of our system design

camera first optimize the wireless cellular network performance in-device and may adapt the video or image stream quality to balance the detection accuracy, recognition accuracy and latency to maximize the advertising revenue, and feedbacks the runtime wireless network information to the server. The advertising server may profile the video or image configurations under various wireless and detection/recognition models with offline experiments, selects the optimal video or image quality configuration for the IoT camera, and optionally refine the video or image adaptation based on the vehicle speed recognition and online profiling.

My Contribution. I worked on this work from Nov 2018 to August 2019. In this work, I developed many platforms and finally we decide to apply my platform on the context of adaptive advertisement. Then Ph.D. candidate Zhaowei Tan finished the report and Jinghao Zhao helped me with experiments.

Implementation. There are many programming languages used on establishing the platform.

(1) **Python**: on deep learning models, such as detecting the car and pushing recommendations
(2) **Notejs**: on server side RTSP/RTMP receiver
(3) **Android / Java**: on Advertisement display / local profiling and optimization. This part we deploy on Android because most of the boards available are on Android and also we can make use of MobileInsight.

4 IMPACT-ORIENTED RESEARCH

4.1 Lyme Disease Data Analytics

Motivation. Lyme disease is greatly under-studied; however, patients have banded together to pool their data via the MyLymeData survey through LymeDisease.org to facilitate research. In our project, we apply a variety of classification methods on the MyLymeData dataset to classify a patient's Wellness and generate treatment recommendations for Unwell patients. The suggestions include antibiotics to take, alternative treatments to try, and potential side effects of treatments. With these models, we hope to educate clinicians and patients more on antibiotics, with the goal of improving patient outcomes.

Overall Design. The recommender network is trained on data from a group of patients with known responses and predicts for patients who either have unknown or suboptimal responses. We want to use the patterns of Well patients to predict the choice of antibiotics for Unwell patients. In this way, we base our recommendations for Unwell patients on the best prescription of similar Well patients. First, we train a classifier on Well patients with baseline answers and antibiotic prescription as features and target respectively. Next, we evaluate the effectiveness of this classifier on Well patients. Then, we apply this classifier to Unwell patients. Finally, we compare our model's output with the original Unwell patients' answers. These comparisons can tell us what improvements can be made to Unwell patients' antibiotics as well as provide a sense of how well the framework works.

Antibiotics	Original Patient Data	Model Prediction	Interpretation
Alinia	1	1	Matched
Amoxicillin	1	1	Matched
Amoxicillin Clavulanate	-1	0	Originally unanswered, predict it might not work
Biaxin	-1	1	Originally unanswered, predict it might be *beneficial*
Cedax	-1	0	Originally unanswered, predict it might not work
Cefuroxime	0	0	Matched

Table 2. Interpreting Antibiotic Recommendation (Synthetic Data). The randomly generated patient data in this example matches the recommendation for three of the six selected antibiotics. For the three antibiotics the patient did not give a response for in the survey, the model predicts which will and will not be beneficial for the patient based on similar Well patients. A patient could then take this information and ask a doctor for more information about the recommended antibiotics.

We trained a fully connected neural network (FCNN) model with the Well patients' baseline questions as input. The outputs are their responses to antibiotic related questions (embedded as one-hot vectors). An example of the output from this model is given in Table 2. The 10-fold cross-validation score on Well group is 0.96, which indicates that the model is trained successfully.

This framework can offer us insights on the redistribution of antibiotics. We use a state map to visualize suggested changes. We compare the original data from the survey and the predicted data from our framework by subtracting the original from the predicted value and storing it in a matrix. We then plot the difference matrix onto a state map. Each state map illustrates a potential redistribution of an antibiotic. States are colored red to indicate that our framework recommends fewer patients to take the selected antibiotic than

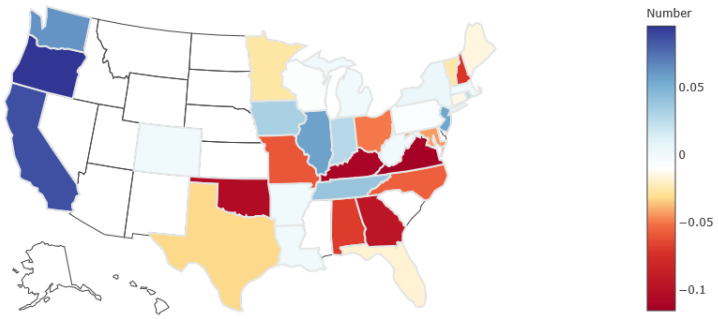

Fig. 14. State map of the oral antibiotic Minocin (Minocycline) showing the percentage difference between actual data and prediction with our model. Only states with more than 13 patients in the sample are colored. This graph illustrates a potential redistribution of the antibiotic. Red color means fewer predictions, while blue suggests more. A darker color corresponds to a larger percentage difference in a state.

are currently taking it in that state, while states are colored blue to indicate more patients are recommended to take that antibiotic.

My Contribution. This work is done during my REU in UCLA Mathematics department under supervision of Prof. Deanna Needell. During the 8-week REU in 2018, I designed and implemented this recommender system mentioned above. Our classification team's work as a whole was presented in JMM 2019 and had real-world deployments with LymeDisease.org.

URL. The paper URL is on http://keplerc.com/publications/JMM2019/JMM_Paper.pdf and the poster URL is on http://keplerc.com/publications/JMM2019/JMM%20Poster.pdf

Implementation. The dataset consists of survey answers from 440 Well and 3686 Unwell patients describing those who have and have not recovered from Lyme disease. These are distinguished by answers to a self-identification question in the survey. We assume self-identification is accurate. These Well and Unwell sets are further divided into Baseline and Phase 1 questions. Baseline questions are identical for both Well and Unwell, while Phase 1 is tailored to either Well or Unwell. In each of these four subgroups, the data is finally divided by data type: binary, categorical, scalar, and string. We study only binary, categorical, and scalar data. Rows correspond to patients, while columns correspond to questions.

We implemented all the classification code in Python, R and matlab.

Deployment. Since this project is directly funded by NSF and LymeDisease.org, the result of our analysis and the recommender framework is directly used by the company myLymeData in order to better understand the patients and to provide valuable insights to patients. The active user for this website is more than 10,000 Lyme disease patients.

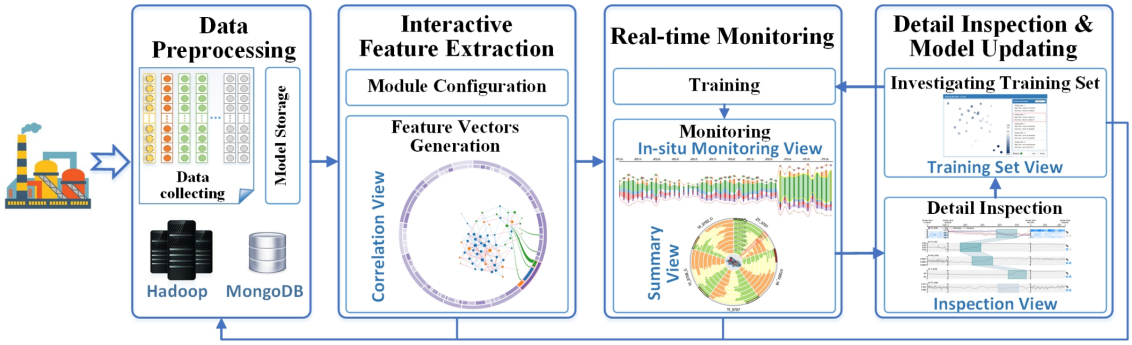

Fig. 15. System workflow: After data preprocessing and interactive feature extraction/configuration, our system can support real-time monitoring and detail inspection and model updating

4.2 Industrial data anomaly detection

Motivation. Monitoring equipment conditions is of great value in manufacturing, which can not only reduce unplanned downtime by early detecting anomalies of equipment but also avoid unnecessary routine maintenance. With the coming era of Industry 4.0 (or industrial internet), more and more assets and machines in plants are equipped with various sensors and information systems, which brings an unprecedented opportunity to capture large-scale and fine-grained data for effective on-line equipment condition monitoring. However, due to the lack of systematic methods, analysts still find it challenging to carry out efficient analyses and extract valuable information from the mass volume of data collected, especially for process industry (e.g., a petrochemical plant) with complex manufacturing procedures. In this paper, we report the design and implementation of an interactive visual analytics system, which helps managers and operators at manufacturing sites leverage their domain knowledge and apply substantial human judgements to guide the automated analytical approaches, thus generating understandable and trustable results for real-world applications.

Overall Design. We design and implement a visual analytics system with a semi-supervised framework to address the major challenges of equipment condition monitoring met by real world operators and managers from a factory of process industry (i.e., a petrochemical plant).Our system integrates advanced analytical algorithms (e.g., Gaussian mixture model with a Bayesian framework) and intuitive visualization designs to provide a comprehensive and adaptive semi-supervised solution to equipment condition monitoring. The example use cases based on a real-world manufacturing dataset and interviews with domain experts demonstrate the effectiveness of our system.

My Contribution. This work is done during my Internship in Siemens Inc. Supervised by Dr. Wenchao Wu, we designed this system together. I surveyed all the literature in anomaly detection and implemented the learning module of the system. Dr. Wenchao designed the visualization part of the paper.

URL. This work is published in IEEE PacificVis 2018 in Kobe, Japan. The URL is on http://keplerc.com/publications/PacificVis2019/PacificVis.pdf.

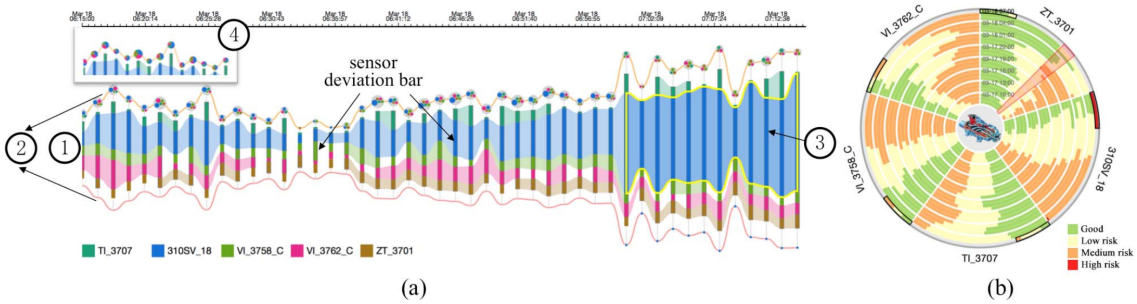

Fig. 16. In-situ Monitoring View shows the risks detected by the monitoring algorithm for all target sensors of a module in real-time; (b) Summary View provides an overview of long-term trends of equipment conditions, which also keeps updating in real-time

Deployment. The system is deployed in Qingdao China Sinopec Petrochemical Plant with a scope of monitoring 3000 sensors.

5 MEDICAL IMAGING

5.1 MRI Skull Stripping

Context. Skull stripping is an important preprocessing step on cerebral Magnetic Resonance (MR) images because unnecessary brain structures, like eye balls and muscles, greatly hinder the accuracy of further automatic diagnosis. To extract important brain tissue quickly, we developed a model named Confidence Segmentation Convolutional Neural Network (CSCNet). CSCNet takes the form of a Fully Convolutional Network (FCN) that adopts an encoder-decoder architecture which gives a reconstructed bitmask with pixel-wise confidence level. During our experiments, a crossvalidation was performed on 750 MRI slices of the brain and demonstrated the high accuracy of the model (dice score: 0.97 ± 0.005) with a prediction time of less than 0.5 seconds

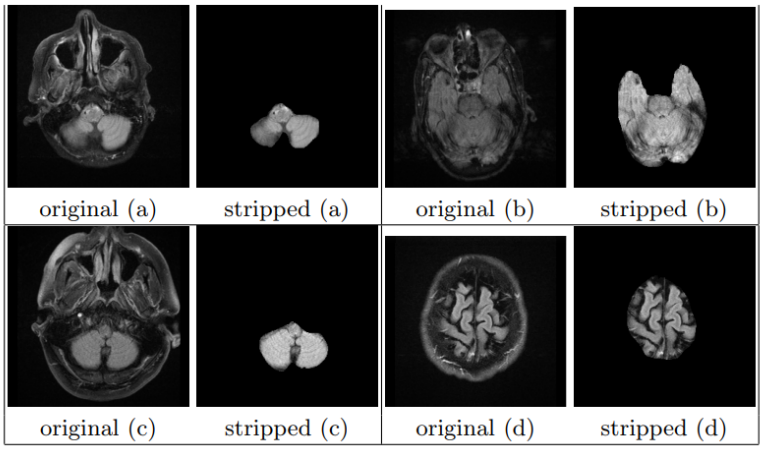

Fig. 17. Results of our proposed Skull Stripping method

Overall Design. We propose an Encoder-decoder Architecture that handles pixel-wise labeling, most Fully Convolutional Network (FCN)s adopt an encoder-decoder scheme. FCN is translation invariant since it leverages operations like convolution, pooling and activation functions on relative distance space. Under this encoder-decoder scheme, input images first go through convolutions and pooling layers to reduce to a more compact low dimensional embedding, and then the target images are reconstructed by deconvolutions and upsampling. Because of the characteristics of FCN, models are usually trained in an end-toend fashion by back-propagation and weights are updated by Stochastic Gradient Descent (SGD) efficiently

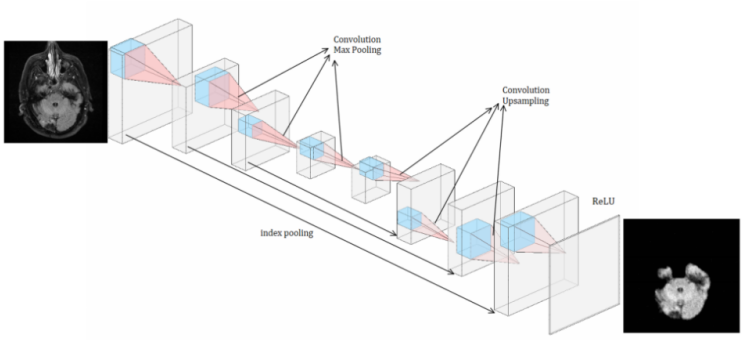

Fig. 18. Proposed Architecture of our CSCNet

My Contribution. I designed and implemented this CSCNet under supervision of Professor Fabein Scalzo. Jingyue Shen was my collaborator who performed all the experiments.

URL. The paper URL is on http://keplerc.com/publications/ISVC/CSC.pdf