Using Temporal Discovery and Data-driven Journey-maps to Predict Customer Satisfaction

Joe Bockhorst*, Yingjian Wang†, Sukrat Gupta‡, Maleeha Qazi§, Mingju Sun¶ and Glenn Fung‖
American Family Insurance
6000 American Parkway, Madison, WI, USA
Emails:*jbockhor,†ywang1,‡sgupta,§mqazi,¶msun, ‖gfung@amfam.com,

***Abstract*—Timely identification of potentially dissatisfied customers enables us to take meaningful interventions to improve customer experience. The goal of this work is to create models that can predict customer satisfaction for active insurance claims at any point in time during the claim process. In order to capture relevant temporal information, we introduce the concept of a "journey-map": a data-driven structured timeline where all the relevant events pertinent to the claim process are registered and positioned temporally with respect to each other. We also describe a machine-learning-based framework to extract and discover meaningful information relevant for the task at hand. The result of this work is a deployed system currently used during the claims process.**

I. Introduction

In today's competitive, consumer-driven marketplace accurate real-time measurements of customer satisfaction is a key focus of many businesses. Here we describe the design and implementation of a recently deployed machine-learning-based system for real-time prediction of customer satisfaction during the insurance claim process (See Figure 1). Accurate predictions would enable proactive interventions and may possibly point out issues related to the claims process.

An insurance policy holder wishing to be reimbursed for a loss files a claim with the insurance company. A claim can be viewed as a series of events, for example, interactions between the insurance company and the insured, that begins with first notice of loss (FNOL) and ends with a final payment. To measure customers' satisfaction insurance companies typically conduct customer surveys after the claim has closed. There is an industry-wide consensus that customer satisfaction is significantly correlated to customer attrition. The more satisfied customers are with the claim process the more likely they will keep their business with the insurance company. The main objective of this paper is two-fold: (a) to describe a machine-learning-based deployed system is that is currently used to predict customer satisfaction (Touchpoint score survey) and facilitate timely interventions during the claim process. (b) to share the lessons learn through the system design process. As part of this work we introduce the concept of a claim process *journey-map*: a data-driven structured timeline where events pertinent to the claim are registered and positioned temporally with respect to each other. The journey maps we use integrate events from multiple heterogeneous data sources. After the customer journey-map representation is created, extraction and discovery of meaningful information relevant to the task

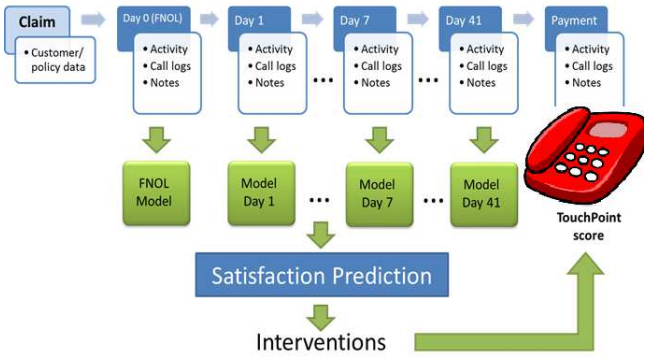

Fig. 1. Our deployed system is used to predict customer satisfaction (Touchpoint score survey) and facilitate timely interventions during the claim process

at hand (prediction of customer satisfaction) is needed. In order to achieve this, we have extended the concept of bag-of-words features used in text mining (NLP) and imaging problems to the temporal domain. Furthermore, the concept of automatically extracting information from journey-maps can be generalized to many processes beyond the insurance industry e.g.: customer lifetime chronology, patient medical history, customer web interactions. We believe it's an area with great potential and applications in several other domains. In summary our discovery and system design process can be divided into three main stages:

1) **Processing and extraction of raw data**: Raw data is collected from three primary sources of the claim process: loss (accident) details, claim notes, and call and activity logs.
2) **Data blending and representation**: Data is combined to derive a data-driven customer journey map; from which predictive composite patterns are derived through machine learning techniques
3) **Feature Discovery and model design:** Discovered relevant features and patterns are used to create models to predict customer claim satisfaction

The rest of the paper is presented as follows: To start with, we present some related work in section II, after that in section III, we describe the data used for this project and how we generated and extracted features from the data-

driven journey-maps. Then in section IV we provide a general description of the deployed system and the predictive models created. In section V we briefly mention some key aspects of the architecture used for system operationalization and the reasons behind these choices. To end the paper we share some empirical results, conclusions and future work in sections VI and VII.

II. Related Work

In recent years, there has been increased interest in various industrial domains for data mining/machine-learning-based systems to predict customer satisfaction. Applications range from predicting fast-food restaurant customer satisfaction [12] to real-time measurement of customer satisfaction after an operator attended call [10], [4]. However, most of these systems are exclusively developed by private companies, hence formal publicly available documentation is scarce.

One of the unique characteristics of our work presented here, is that we aim to predict customer satisfaction at anytime during a complex temporal process that comprises events produced and extracted from several heterogeneous systems and data sources. In a sense, our created data-driven journey-maps not only capture and consider the path of the customer during the claim process but also the underlying operations, events, and transactions that occur without the customer's awareness.

The work presented here can be generalized as a methodology for discovering information from any customer-related temporal process and link it to any measure of interest (satisfaction, retention, etc.).

There is few prior work that address settings closely related to this general framework. In [3], a reinforced learning method is proposed to learn patterns from partial interaction sequences so information acquired from customers can be efficiently assimilated and applied in subsequent interactions with other customers. However this methodology requires well-defined sequence outputs or feedback that can be used as rewards (labels), which is not the case in our setting, since we only have customer feedback at the end of the process.

Regarding the concept of journey-maps, even though it is a "hot" industrial topic, there are not many papers about it in the machine learning/data mining literature (that we know of). Most of the publications we have seen are white papers from private companies that introduce the journey-map concept and discuss its value from a business point of view. An example is [9]. However, there has been an increase in use of the term in different industries, but again no formal generally accepted or academic definition exists.

One of the main characteristics of the system presented here is the ability to discover meaningful event associations hidden in journey-maps based on customer generated data. Most of the existing temporal mining methods [6] are not adequate for such a task. For example, a popular class of algorithms that are variations of the frequent itemsets algorithm [1] are only inspired by customer transactions and focus more on the sequential nature of occurrences and (a) do not handle event repetitions gracefully and (b) only the order among events is considered and not the temporal aspect (how far apart events are from each other). The methodology described in this paper addresses both of these issues.

The T-pattern algorithm [7] was created to detect temporal patterns in human behavior. Characteristics of relevant behavior patterns are identified statistically and combined in order to define a scale-independent, hierarchical time pattern type, called a T-pattern. However, this method was developed for modeling one sample process at a time (not a set of many customer's processes) and does not extend or scale for thousands of customers.

III. Data sources and Representation

There are myriad potential sources of direct or indirect influence that may help explain a customer's satisfaction following a claim. These can be grouped into i) policy data, ii) insured's previous experience with the claim process and the insurance company, iii) insured's expectations, iv) characteristics of the loss or accident, v) actions taken by the claim adjuster or other agents of the insurance company, vi) insured's mood and overall mental state when the survey is conducted, vii) qualities of the representative conducting the survey. Our system considers features for the first five groups. More specifically, primary data sources considered and used by the system are: 1) Policy and incident data, 2) Claim handler notes and 3) Claim process and customer interaction data. A more detailed description of the these data sources is presented next.

A. Policy and Incident Data

We have data collected at the moment the claim is reported by the insured, called FNOL (First Notice of Loss). This data consists of peril codes, loss totals estimates, etc. Additionally we are also considering policy related data: type of coverages, customer history, demographics, etc. We also have access to the satisfaction scores from Touch Point Surveys; a survey sent to customers after the claim is closed to gauge their experience on a 5-point scale. Scores of 1 and 2 are considered "dissatisfied scores" while scores of 3, 4 and 5 are considered "satisfied". The main goal of the project is to predict (at any time during the claim process) whether a customer will be satisfied or not when the claim closes.

In order to identify potentially dissatisfied customers, certain events that occur in the majority of claims were identified by the business to be of particular interest. These events include: FNOL (first notice of loss), adjuster assignment, initial contact with the insured party, inspection completion, estimate completion, payments made to the insured/on behalf of the insured, perils opened and closed on the claim, and if the claim was a storm loss. From these identified events of interest, certain features can be derived that make logical business sense: time between any two events *e.g.*, FNOL to any other event of interest, first contact to inspection, etc.

Due to the nature of the systems that generate data some of these events and/or features can be calculated exactly,

others can only be proxied or approximated using other data generated by the same systems during the claim process.

B. Claim Handler Notes

These are all the notes that capture information gathered during every interaction with the customer, other parties involved in the incident, repair process, etc. Given the various systems used to collect the data, these have a lot of variation, but are all unstructured text. The available claim notes were in raw and unstructured form. we converted all the words in the note to lower case in order to bring uniformity in the notes. After tokenization, the notes were then processed using a part-of-speech (PoS) tagger, which was trained on a Wall Street Journal (WSJ) dataset to identify the most important parts of the note: nouns, adjectives, verbs, and adverbs.

We noticed that notes varied in size drastically. Some notes had only five words, while others had more than a hundred words. This dramatic variation on note size would have led to a very sparse tf-idf matrix representation, which would adversely affect the text score generation. So, in order to ensure that all the notes were getting considered, as well as a denser matrix being generated, we decided to merge the notes in a dataset by claim.

We also wanted to track fluctuation in customers' satisfaction as the claim process progresses, for this we decided to split the dataset based on the timestamp on the notes. We took the date of FNOL as the reference to computed relative day of note creation for the notes for each claim ($rel\_day = note\_date - fnol\_date$)

The dataset was then split into sets of notes for each *rel_day*, each set containing notes created on or before the particular *rel_day*. After experimentation, we decided to generate a tf-idf matrix for the top 2000 more frequent words. We then applied principal component analysis (PCA) in order to achieve a denser and more uncorrelated set of features. After testing on the validation set, we settled on a dimensionality reduction to 500 features.

C. Claim Process and Customer Interaction Data

There are 3 major sources from which our events originate: call logs - the AmFam telephone system that logs customer's calls, claim notes - the claim system notes metadata, and the activity log - an internal log system recording significant activities during the claim process.

For example, an *Inbound phone call* event from the phone system represents an event where a customer calls the AmFam customer hotline; a *Reason for Escalation* event from the claim notes records certain reasons that an escalation of the claim handling is needed; a *Make Payment Submit* event from the activity log indicates that a payment request has been submitted in a response to the customer's insurance coverage request.

For this project, a total of 238 event types are discovered from the 3 sources, among which 2 (*Inbound phone call* and *Outbound phone call*) are from the phone system; 23 events are extracted from the claim notes creation process; and 213 from the activity log from the claims system.

Since the claim process can be represented as a series of events ordered chronologically from the moment that the claim is reported until it closes, it is important to note that a single event type can have multiple occurrences (e.g.: we can have several inbound phone calls during the claim process). Usually the occurrences of different event types alternately appear on the journey-map spawning specific ***patterns*** that capture information about the course and progress of the claim at any moment. This notion inspired us to (a) create a data-driven temporal representation of the claim process (journey-map) and (b) to efficiently retrieve event-related information from the data-driven journey-map and use it for customer satisfaction prediction.

D. Extracting Features from Temporal Journey-maps

In order to extract features from the the data-driven temporal journey-maps, we used a technique inspired by the bag-of-words concept used in information extraction and NLP [2]. The bag-of-words technique consists of representing a text document by vectors that counts word occurrences according to a predefined word dictionary. A similar idea is also used to represent images in computer vision by considering an image as a vector of occurrence counts of a vocabulary of local image features [13]. We can think of a journey-map as a text document where the time events (belonging to a predefined set of events or event dictionary) are the words that occur in the aforementioned document. Figure 2 provides a graphical representation of this notion. Note that analogous to the text bag-of-words representation where grammar and word order are not taken into account, this vector representation does not preserve the order of the events in the original journey-map. However, this kind of representation is often successful in text applications, and it was effective for our problem as well. For the rest of the paper we will refer to this feature set as the "temporal bag-of-events" features or TBOE. In order to account for some of the TBOE representation shortcomings, some extra complementary pieces of information (e.g. the times when the events of this type occur) are collected. Based on the collected information, we retrieve some statistics associated with this event type to create additional features as we explain next.

For all 238 event types, there are 5 basic features: *Events count*, *Earliest occurrence time*, *Latest occurrence time*, *Average occurrence time*, and *Standard deviation of occurrence time*. For the 2 phone call events, there are 4 additional features: *Average duration*, *Maximum of duration*, *Average holding on time*, and *Maximum of holding on time*. Among the 213 event types from the Activity Log system, there are 28 event types which are related to payment. For these 28 event types, there are 3 additional features: *Average payment amount*, *Maximum payment amount*, and *Total payment amount*. Thus from all the 238 event types, we retrieve a total of 1282 features for 166505 claims, which forms a matrix with the size of 166505×1282. We are interested in the variation

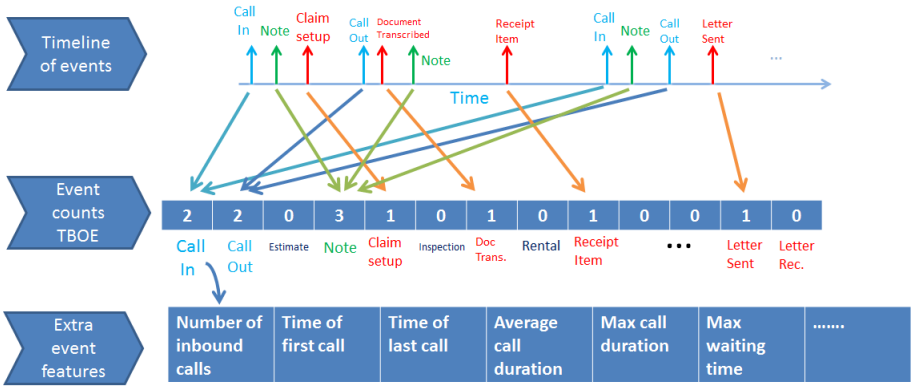

Fig. 2. The Temporal bag-of-words (TBOE) representation

of the degrees of the customers' satisfaction at different time instances across the life of the claims. For example, we want to calculate the satisfaction scores for every day in the first couple of weeks since the opening of a claim. While for the 24 weeks following the first week, we are only interested in the last day of each week (a total of 26 weeks). To this end, features are calculated for 38 different time instances, forming a TBOE tensor, a 3-dimensional matrix with the size of $170160 \times 1282 \times 38$, among which we show experiment results for day 2, 3, 5, 7, 8, 13, and 27. Due to our project scope and timelines, we did not exploited this data structure to its full potential, however, we believe that this data structure will allow us to explore more complex ways to discover knowledge from this data in the future.

IV. Predictive Models

Our aim is to predict customer satisfaction after the claim process has ended while the claim is still ongoing. We have labels from surveys conducted after the claim but we do not have satisfaction observations taken during the claim. This is a challenge for interpretation because on a given day the model may predict that the customer will be dissatisfied at the end of the claim but that customer may be perfectly satisfied at the time. The prediction may be due to factors related to the claim or loss that suggest that claims processes will be followed that have a propensity for dissatisfaction. One approach to prediction on temporal process is to explicitly model dynamic aspects either without any hidden state, such as a Markov chain, or with hidden state, for example hidden Markov models or partially observable Markov decision processes.

Our approach does not utilize one of these dynamic models for a number of reasons. Since we do not have observations of satisfaction other than at the end of the claim we have no labels that we can use the train the system on how satisfaction evolves. Attempting to model the dynamics of the claim process while interesting in its own right, is significantly more complex than the given problem and was prohibitive given the project timelines. Further, previous work suggests that the added modeling complexity is more likely to be less accurate than a direct model. In any case, an overall claims model was not the motivation for the system in the first place.

A. Global vs. Daily Models

One key modeling decision we faced was whether we would learn a single global model (it was clear from the outset that separate models were required for auto and property claims) that we would employ on each day that predictions are desired, or if we would utilize train-separate models for every day we want to have predictions. We opted for training daily models. The primary motivation for our choice was empirical (see Results), the daily models clearly out perform the global, but there is also compelling rationale for our choice. The relationship between features and customer satisfaction are sometimes strongly dependent on the stage of the claims process. For example, consider the variable *number of phone calls*. In the first few days after FNOL this variable is positively correlated with satisfaction as more activity in this early time period suggests that the claim is moving toward a timely close. Later on, however, we find *number of phone calls* to be negatively correlated with satisfaction as a large number of phone calls suggests confusion or complexity, both of which are believed to be causal factors of dissatisfaction. Similar reasoning applies to a number of other features including *estimate complete* and *number of claims notes*.

B. Different Models for Different Time Instances

Each day following the day the claim was reported, or FNOL, for which we wish to have predictions our system trains and uses two binary classification models of satisfaction. The first of these are text-classification models that predict the probability of end-of-claim claimant satisfaction given only the text in the claim notes. We call the probability of satisfaction given the claim notes the *text-score*. The second classifiers are linear support vector machines whose features include the *text-score* (more detail in section IV-C) along with the features computed from the other data sources described before. For example, to predict the end-of-claim satisfaction for a claimant n days after FNOL we use the FNOL+n text model to obtain

the *text-score* and then use the SVM to obtain the probability of end-of-claim dissatisfaction given all our evidence to date.

C. Customer Satisfaction Text Score

We made the decision to include *t*ext score as a feature of our final model rather than have text based features directly included in the final model. The downside of this approach is that it prevented subtle interactions between text and non-text features to be uncovered and utilized for prediction. On the positive side it lends itself well to modularization and clear separation of concerns. Our modeler working on the final model only need be aware that a *t*ext-score was coming and the text modeler could work on getting the best text model without dealing with the additional complexity of 1000s of other features. The driving force behind our choice was that it would be easier for us to explain the model to business if we did not open the hood and give them full view into all the features of the text model. In our environment, the business preference is for a causal mechanism be provided with every variable included in the final model, even for tasks for which the accuracy of the prediction is paramount. The Text Score model that we used in this project is a traditional Natural Language Processing model.Term frequency-inverse document frequency (tf-idf) is used to select the required words by considering the occurrence of a given word in the current document and number of documents in which the word has occurred ([5]). Using the top-n words based on term frequencies as features, we can use maximum information from the claim notes to generate a text score, which, in turn, will act as a feature to predict claim satisfaction score.

In order to generate the *text score* for the models, we decided to use the output of a binary classifier based on the features we generated using tf-idf and PCA ([11]). We decided to focus on linear classifiers (SVM) due to their lower complexity. Using the claim's distance from decision boundary generated by a linear SVM, we generated a *text score* feature, which would be used in conjunction with the other features described above.

V. Architecture and Operationalization

In order to put the satisfaction model output into the business user's hands, the model has to be integrated with operational systems the business uses every day. This process is referred to as model operationalization. Like any software development and delivery process, model operationalization requires architecture and design. In addition to enabling integration between the satisfaction model presented here and the company claims system that business users use every day, the operationalization architecture for the predictive model was designed to have the following distinctive characteristics:

- It allows end-to-end data and model pipelines implementation to be co-located on the Hadoop platform.
- It allows the data and model pipelines to scale out with the distributed computing technique of MapReduce.
- It allows a smooth transition to migrate the satisfaction model from development to the production environment.

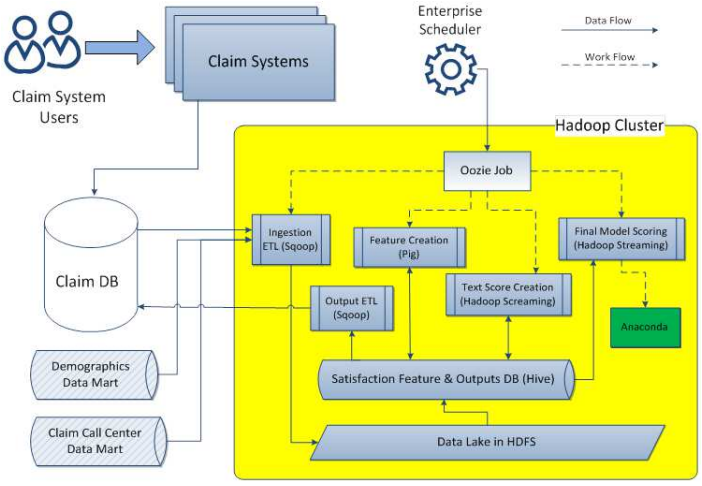

Fig. 3. The architecture of the deployed system

	Auto	Property
Satisfied	105541 (96.8%)	53090 (92.4%)
Dissatisfied	3490 (3.2%)	4384 (7.6%)

TABLE I
COUNTS OF THE NUMBER OF SATISFIED AND DISSATISFIED CLAIMS IN OUR EXPERIMENTAL DATA SETS.

This architecture allowed us to deploy the satisfaction model to production in a shortest time possible with minimal changes to the model codebase. Most importantly it reduced the overall complexity of the model operationalization by centralizing all artifacts on a single Hadoop platform.

Figure 3 shows the conceptual view of the satisfaction model operationalization architecture. It captures the computing platform (Hadoop), data sources (Operational System back-end in the form of database and data marts in the enterprise data warehouse), system integration points, and major processing components.

VI. Results

Here we report findings from experiments we have conducted in order to assess various aspects of our system. We report results for seven days of interest to our partners from the claims division: 2, 3, 5, 7, 8, 13 and 27 days following FNOL. We refer to this set of days as *prediction days*.

A. Methodology

Our experiments use a data set of 166,505 claims between January 2009 and August 2014 for which we have Touch Point survey responses taken within a week of claim closure. Our aim is to model responses to the question: "*Please rate your overall experience with American Family on your most recent claim. Would you say it was Excellent, Above Average, Average, Below Average, Poor?*"

Customer responses are encoded on a scale from 1 (Poor) to 5 (Excellent) and we formulate our problem as the binary classification task of distinguishing dissatisfied responses (Below Average and Poor) from satisfied responses (Average,

Above Average and Excellent). This problem is characterized by high skew as satisfied responses far out number dissatisfied ones (see Table I). Because the claims process along with the suspected causes and correlates of customer satisfaction are quite different between auto and property claims we learn separate models for each. For simplicity in this section we focus our discussion on a a single type of claim but it should be understood that all steps (feature selection, parameter tuning, *etc*.) are done separately for auto and property.

1) Feature Selection and Parameter Tuning: For practical considerations including simplicity of implementation and communication across business units we use feature selection to identify a small subset of the 2,249 candidate features (after one-hot encoding of nominal features) to use in the deployed model. While performing feature selection separately for each *prediction day* may lead to more accurate models, we expect this effect to be slight. After all, if a feature is important for prediction a certain number of days following FNOL we expect it likely to have value other days as well. For this reason along with our bias towards simplicity and the realities of time constraints we performed feature selection jointly for all *prediction days*. We perform greedy forward feature selection optimizing for the area under the ROC curve (AUC) using ten-fold cross validation on our training sets. Our deployed models, whose results we report here, have 44 features for auto claims and 42 features for property claims. We observe little improvement in AUC for additional features. We tune the SVM regularization parameter C separately for each day. We consider $C = 2^n$ for n equal to the integers between -11 and 2 inclusive and choose the value that maximizes mean accuracy on a ten fold cross-validation experiment on the training set.

2) Missing Values: Dealing with missing values is an important practical consideration as a number of key features were potentially missing. The LinearSVC classification models in scikit-learn do not support missing values, so we needed to come up with a custom approach. We treat missing values in nominal and numeric features differently. For nominal features along with one-hot encoding and we also introduce a special *_is_NA* feature that is one when the corresponding feature is missing and zero otherwise. For example, for the feature *color* with the three possible values red, blue and green we would create the four binary features *color_is_red*, *color_is_blue*, *color_is_green* and *color_is_NA*. Dealing with continuous features presented a greater challenge as the simple approach of imputing missing values with the training set mean was not a good fit given the reasons for missing values in our domain. The aggregate timeline features, for instance, are undefined whenever the count for the corresponding event type is zero. When a claim has not had any outbound phone calls filling in the value of *max_time_of_outbound_phone_call* within the training set mean does not make sense. Instead, we convert numeric features to nominal features using binning and again introduce the special *_is_NA* feature just as we do for naturally nominal features. We set bin boundaries so that a (nearly) equal number of points fall in each bin. We limit continuous features to at most ten bins.

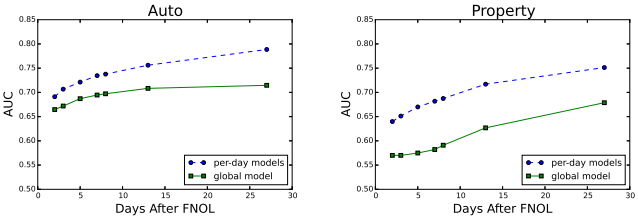

Fig. 4. Comparison of AUC score of the global model versus per-day models for the property text model (right) and the auto text model (left).

3) Global Models: Although our design bias was toward simplicity we suspected that using separate models for each *prediction day* would yield more accurate predictions (see Section IV-C). To determine if the costs of the increased complexity were justified we compared the accuracy of the per-day models with a single global model. We trained global models for both types of claims (auto and property) separately. To train the model for one claim type, we pooled together training examples from all *prediction days* of that type into a single training set. The pooling was done following feature selection.

4) Final Model Results: Prior to deployment we conducted an evaluation using our held aside test set to estimate what the performance of our models would be if deployed. Figure 4 shows plots of test-set AUC scores at the *prediction days* For the global model and the per-day models. These plots reveal two key findings. First, accuracy improves with increasing number of days after FNOL. This is not a big surprise, since the amount of information available to inform predictions increases with time. However, the strength of this trend suggests that the causes of dissatisfaction include events that occur during the claim process that cannot be predicted exactly only from information available at FNOL such as policy details and attributes of the loss. On the other hand, that there is predictive value in models only 2 days after FNOL indicates that some aspects of dissatisfaction are known at FNOL. The second key finding from Figure 4 is that the per-day models clearly outperform the global model. The per-day model has a higher AUC than the global model at all time-points. For property claims the per-day model has a relatively constant boost over the global model of about 0.07 while for auto claims the difference starts out quite small, about 0.025 at day 2, but increases steadily over time.

B. Post-deployment Results

Experiments using held aside test sets are important to estimate the post-deployment performance of the system. Such results are, however, not statistical guarantees of future performance levels for a number of reasons such as i) unknown biases in the training / test set instances, ii) future changes in the business processes generating the data, iii) previous changes in the same business practices, leading to biases in the training / test sets, iv) sampling effects, and many more. For these reasons it is essential to periodically assess the

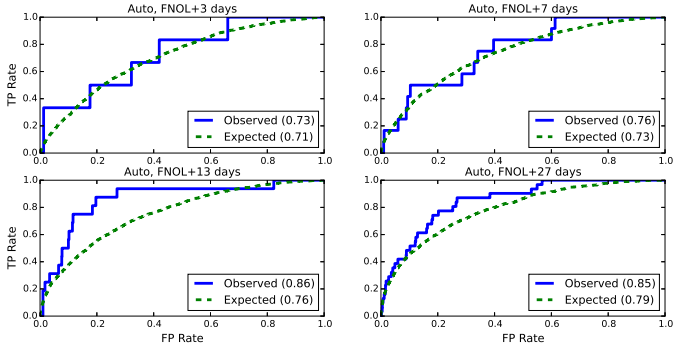

Fig. 5. ROC curves of trained models on our test set (Expected) and post-production data set (Observed). The area under the ROC is given in parentheses.

performance of the deployed system and compare it to the expected performance. We have performed one such post-deployment evaluation approximately 60 days after deployment, at which time we had over 400 surveyed claims of both auto and property. Figure 5 shows the observed (post-deployment) and expected (test-set) ROC curves. For these plots dissatisfied responses are positive examples and satisfied responses are negative examples. These plots show that the performance of the deployed system is close to expected. If anything, performance of the deployed system has been slightly better than expected thus far. This performance test was repeated 10 months from the date of deployment and the results obtained are similar to the ones obtained for the 60 days test.

VII. Conclusions and Future Work

In this work we described a system for predicting whether or not an insurance customer will be satisfied at the conclusion of a claim the customer has opened. The attributes the system uses to predict satisfaction are derived from multiple data sources including: details of the reported loss or accident, patterns of telephone calls between the customer and the insurance company, free text notes entered into internal systems by claims adjusters among others.

To incorporate the heterogeneous and variable-length collection of time-stamped events, claim notes, and phone calls we developed an approach for constructing a fixed-size set of TBOE (*temporal bag of events*) features for each claim. This method was inspired by and is similar to the *bag-of-words* features popular in natural language and image processing.

The system has been implemented for the production environment, has been deployed, and is currently used each day to predict the end-of-claim satisfaction level of all customers with ongoing auto or property claims with American Family Insurance. A sixty day post-deployment evaluation shows that the deployed system is performing as expected according our estimates based on held aside test sets. We want to further explore the concept of temporal bag-of-words features (TBOW) including the concept of bi-grams and n-grams to consider event order and interactions in a more complex fashion. We are currently actively researching how to extend the main ideas behind the T-pattern algorithm [7] to consider a set of timelines simultaneously instead of only one at the time. In addition, we are exploring ways to make the modified T-pattern algorithm scalable so it can be used in the context of big data. We are also exploring incorporation of newly available data sources like speech-to-text transcription of phone calls, call sentiment, and more complex call statistics like number of silences in a call, voice emotion, etc.

VIII. Acknowledgments

The success of this project required a lot of guidance and assistance from many people and we were extremely fortunate to have gotten this all along the way. We would like to express our sincere appreciation to the various American Family Insurance teams & personnel who helped make this work possible: the claims team, personal lines (esp. Sandra Muller for getting us started down the right path), the Strategic Data & Analytics (SD&A) data team for gathering data from various sources, and the SD&A management team. We would also like to thank the I/S team, without whom the implementation and deployment of the final model for the benefit of business would not be possible.

References

[1] R. G. A. Tiwari and D. Agrawal. A survey on frequent pattern mining: Current status and challenging issues. *Information Technology Journal*, 9:1278–1293, 2010.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, NY, 1999.

[3] L. N. e. a. David Silver. Concurrent reinforcement learning from customer interactions. *JMLR*, 2013(28):924932, 2012.

[4] F. P. e. a. Ernesto Diaz-Aviles. Towards real-time customer experience prediction for telecommunication operators. *CoRR*, abs/1508.02884, 2015.

[5] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall, New Jersey, second edition, 2008.

[6] S. Laxman and P. S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.

[7] M. S. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments, & Computers*, 32(1):93–110, 2000.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[9] H. of the customer. Customer journey map white paper. *http://www.heartofthecustomer.com/customer-journey-map-white-paper/*, 2013.

[10] Y. Park and S. C. Gates. Towards real-time measurement of customer satisfaction using automatically generated call transcripts. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1387–1396. ACM, 2009.

[11] F. e. a. Pedregosa. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] B. A. Tama. Data mining for predicting customer satisfaction in fast-food restaurant. *Journal of Theoretical and Applied Information Technology*, 75(1), May 2015.

[13] C.-F. Tsai. Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012(376804):19, 2012.