ACCESS SOFTEK, INC

DEEP LEARNING WITH MICROSOFT COGNITIVE TOOLKIT CNTK IN REAL LIFE PROJECTS

LIUBOV KAPUSTINA, DATA SCIENTIST

IGOR YAROSHENKO, DATA SCIENTIST

AI UKRAINE

13-14 October, 2018

LIUBOV KAPUSTINA, DATA SCIENTIST

Liubov Kapustina is a data scientist with more than 10 years' experience in the industry. She has experience in risk scores and predictive modeling in Banks and in consulting. She has wide experience of realized real production projects with AI and Data science. She also participated in the Marketing Revolution 2015 and AI Ukraine 2015 conferences as a speaker. She is member and speaker in Kiev Big Data Community.

https://www.linkedin.com/in/liubov-k-03164525/

IGOR YAROSHENKO, DATA SCIENTIST

ACCESS SOFTEK, INC

Igor Yaroshenko is young and ambitious data scientist with strong technical background. He has experience in bank industry and e-commerce projects. He has wide experience of realized real production projects with AI and Data science. Igor has participated in the conference Nordic Business Day 2016 as a speaker with talk “Data Science innovation in Agriculture (smart hive)” and Kyivstar Big Data Hackathon.

https://www.linkedin.com/in/igor-yaroshenko-0468b369/

PLAN

Part I: how we choose CNTK + c#

- Access Softek company, our tasks, data volumes
- Comparison CNTK with other existing solutions
- 3 pillars of our requirements: speed, fast and cheap training and accuracy

Part II: CNTK + c# in real life projects

- Fraud detection
- User Engagement
- Churn Prevention

PLAN

Part I: how we choose CNTK + c#

- **Access Softek company, our tasks, data volumes**
- Comparison CNTK with other existing solutions
- 3 pillars of our requirements: speed, fast and cheap training and accuracy

Part II: CNTK + c# in real life projects

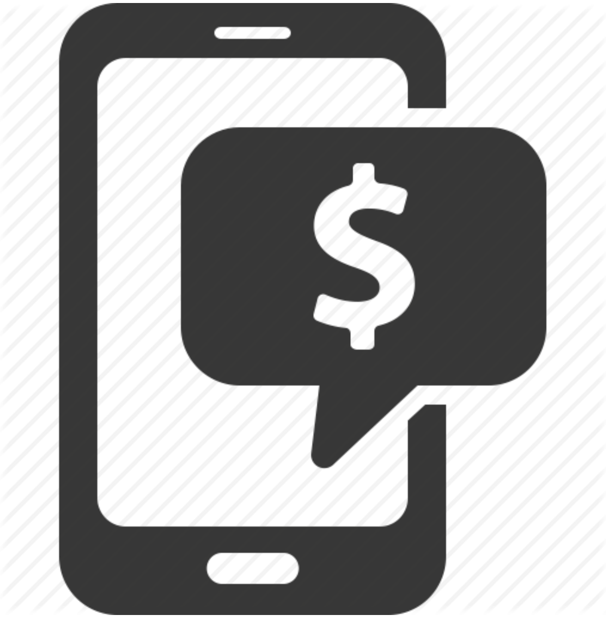
- Fraud detection
- User Engagement
- Churn Prevention

ACCESS SOFTEK IS MOBILE-FIRST AND A.I. EVERYWHERE

ACCESS SOFTEK, INC

mobile banking

online banking



Over **3 million** active users per month

400+ financial institution clients

PLAN

ACCESS SOFTEK, INC

Part I: how we choose CNTK + c#

- Access Softek company, our tasks, data volumes
- **3 pillars of our requirements: speed, fast and cheap training and accuracy**
- Comparison CNTK with other existing solutions

Part II: CNTK + c# in real life projects

- Fraud detection
- User Engagement
- Churn Prevention

SPEED, FAST AND CHEAP TRAINING AND ACCURACY

Why are these requirements so important?

Firstly, the user must get permission or be prohibited from the transaction in real-time.
For a backend serving requests from multiple clients, C# with just-in-time compilation is incomparably more efficient than Python.

Secondly, there are very few examples of fraud, and they are often unique. It's like a flu virus: once we have taught the model to define one type, a new type can immediately appear.

How quickly should the model learn when new data is available?
As soon as possible!

Microsoft Cognitive Toolkit library trains LSTM-models up to 4 times faster than TensorFlow, it parallelizes the processes and uses the server's resources more effectively and efficiently. In this way we can train the model faster and cheaper.

Under The Nilson Reporter, in 2017, for every $100 spent, 7.2 cents is lost due to fraud, therefore the fraud prevention cannot cost more than this, otherwise it will become more expensive than the fraud itself.

Thirdly, the accuracy of TensorFlow and CNTK backends are similar across all benchmarks, and in some of it CNTK is better.

PLAN

Part I: how we choose CNTK + c#

- Access Softek company, our tasks, data volumes
- 3 pillars of our requirements: speed, fast and cheap training and accuracy
- **Comparison CNTK with other existing solutions**

Part II: CNTK + c# in real life projects

- Fraud detection
- User Engagement
- Churn Prevention

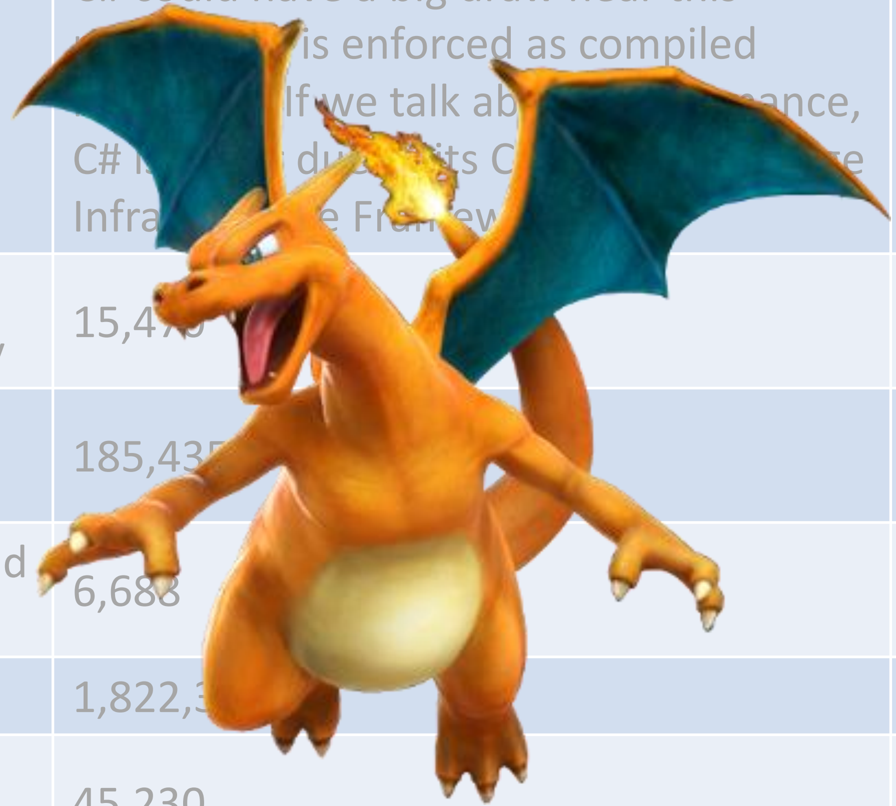
SPEED, FAST AND CHEAP TRAINING AND ACCURACY

** The latest (16th) Techempower benchmarks*	Microsoft .NET *responses per second*	python *responses per second*
Performance	C# could have a big draw near this respect. C# is enforced as compiled language. If we talk about performance, C# is faster due to its Common Language Infrastructure Framework.	Python is enforced as associate taken the language. With Python implementation of the JIT compiler, Python's program execution is improved.
Single query: database-access responses per second, single query	15,470	4,398
Multiple queries: Responses per second at 20 queries per request,	420,820	185,435
Data updates: responses per second at 20 updates per request	6,688	4,390
Plaintext responses per second	1,822,366	975,225
Fortunes, which is a test of read performance	45,230	20,414

SPEED, FAST AND CHEAP TRAINING AND ACCURACY

ACCESS SOFTEK, INC

** The latest (16th) Techempower benchmarks*	Microsoft .NET *responses per second*	python™ *responses per second*
Performance	C# could have a big draw near this [illegible] is enforced as compiled [illegible] If we talk ab[illegible]ance, C# i[illegible] du[illegible] its C[illegible]e Infra[illegible]e Framew[illegible]	Python is enforced as associate taken the language. With Python implementation of the JIT compiler, Pyt[illegible]ecution is improved.
Single query: database-access responses per second, single query	15,4[illegible]	4,398
Multiple queries: Responses per second at 20 queries per request,	185,43[illegible]	420,8[illegible]
Data updates: responses per second at 20 updates per request	6,688	4,39[illegible]
Plaintext responses per second	1,822,3[illegible]	975,225
Fortunes, which is a test of read performance	45,230	20,414



PLAN

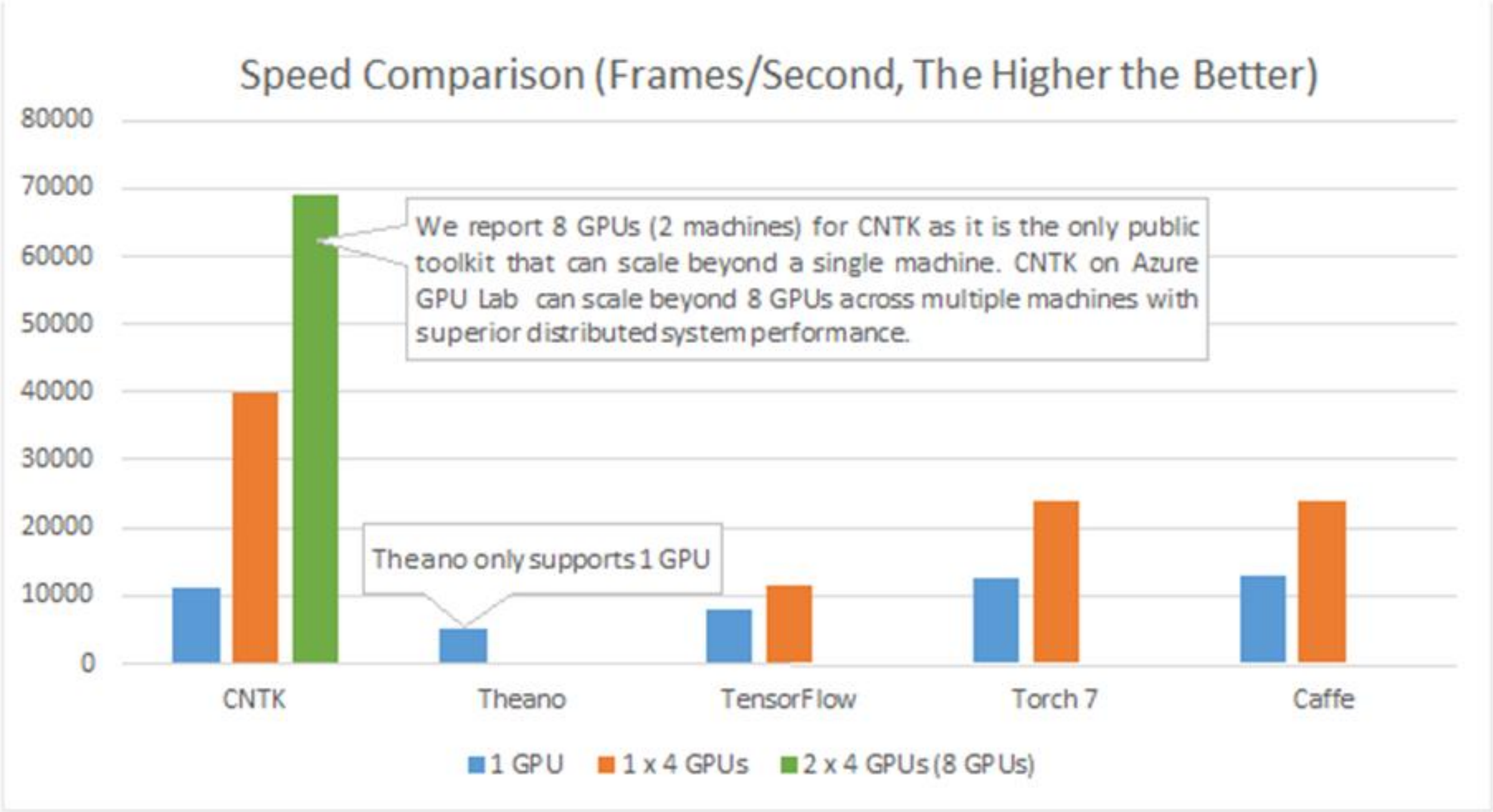
Part I: how we choose CNTK + c#

- Access Softek company, our tasks, data volumes
- 3 pillars of our requirements: speed, fast and cheap training and accuracy
- **Comparison CNTK with other existing solutions**

Part II: CNTK + c# in real life projects

- Fraud detection
- User Engagement
- Churn Prevention

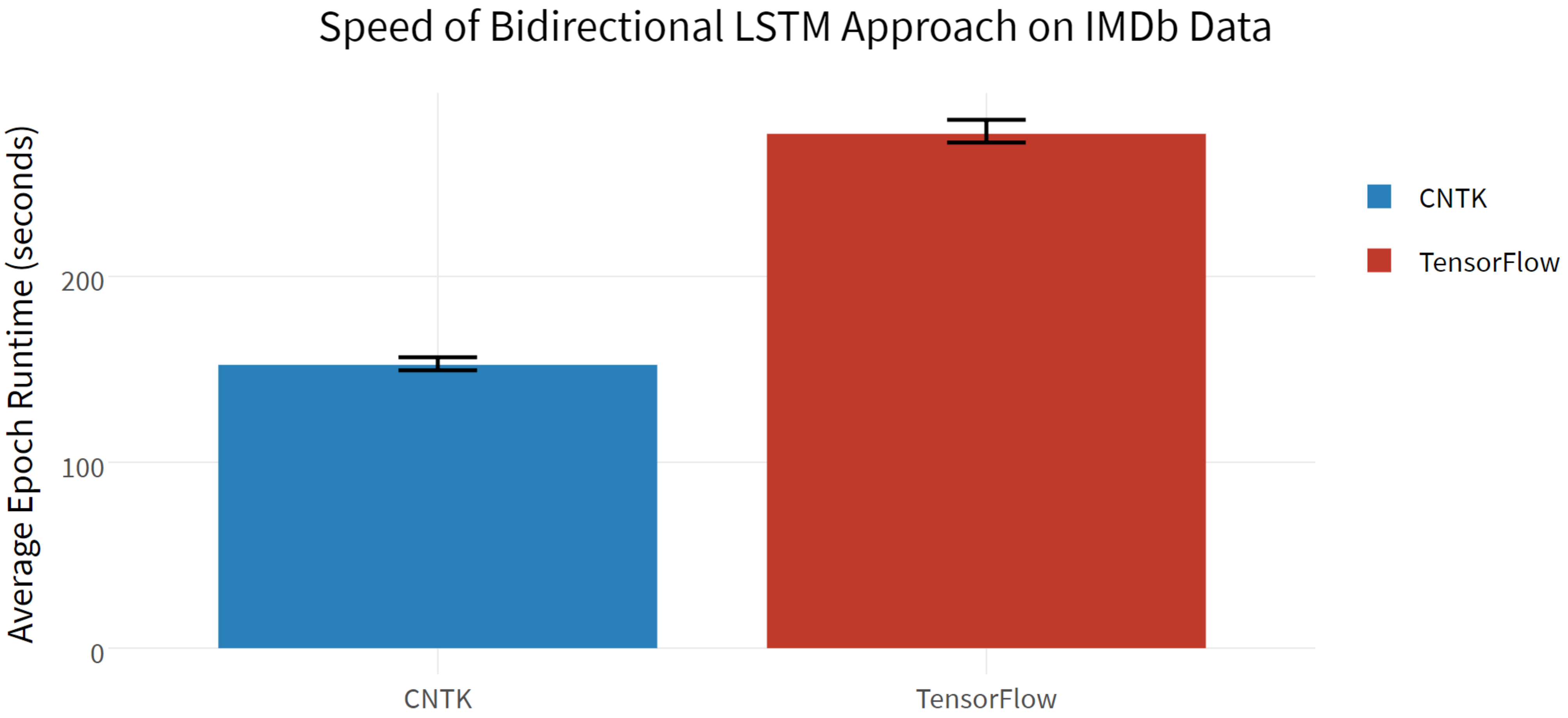
SPEED, FAST AND CHEAP TRAINING AND ACCURACY



SPEED, FAST AND CHEAP TRAINING AND ACCURACY

ACCESS SOFTEK, INC

CNTK IS MUCH FASTER!



PLAN

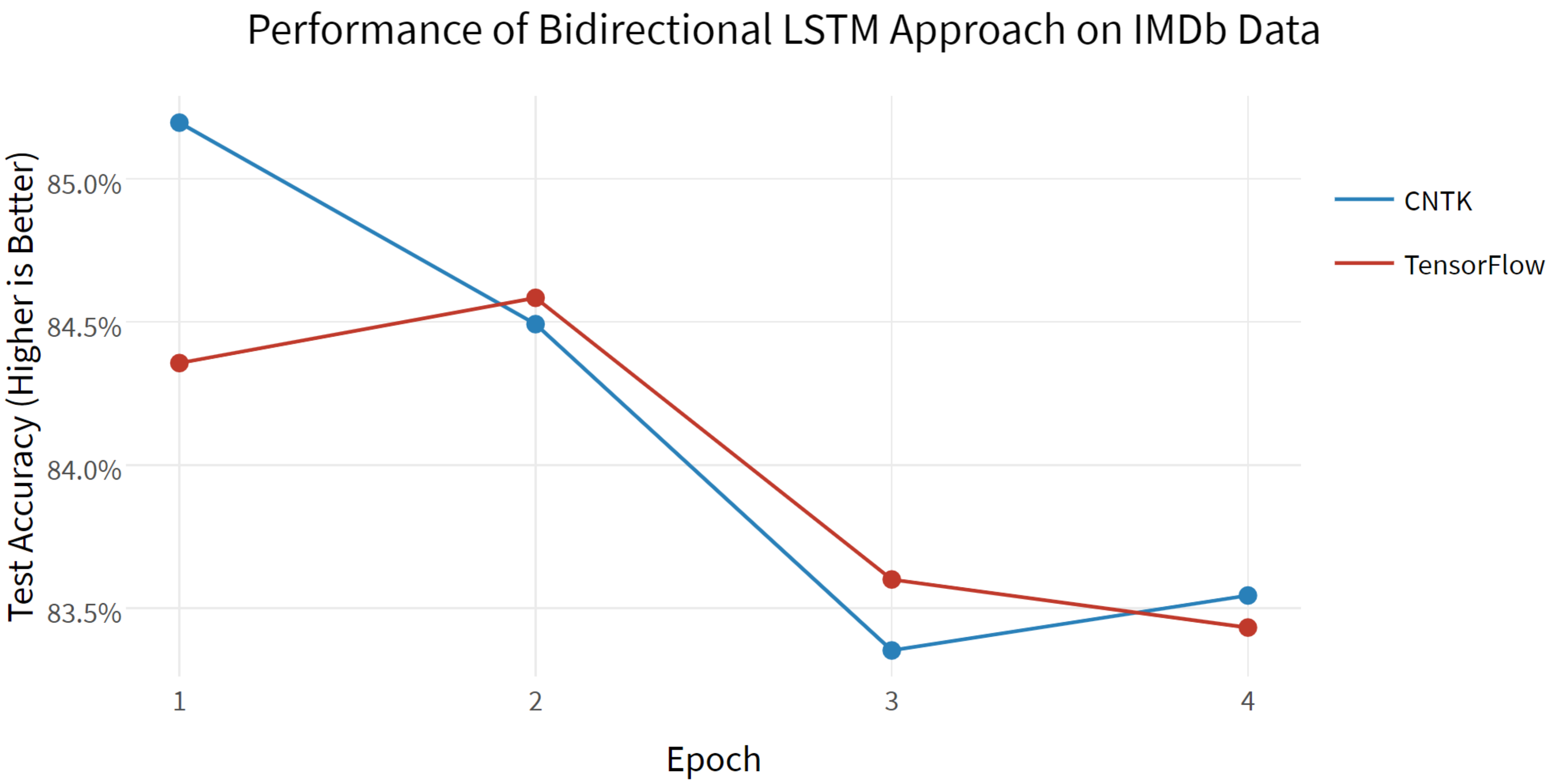
Part I: how we choose CNTK + c#

- Access Softek company, our tasks, data volumes
- 3 pillars of our requirements: speed, fast and cheap training and accuracy
- **Comparison CNTK with other existing solutions**

Part II: CNTK + c# in real life projects

- Fraud detection
- User Engagement
- Churn Prevention

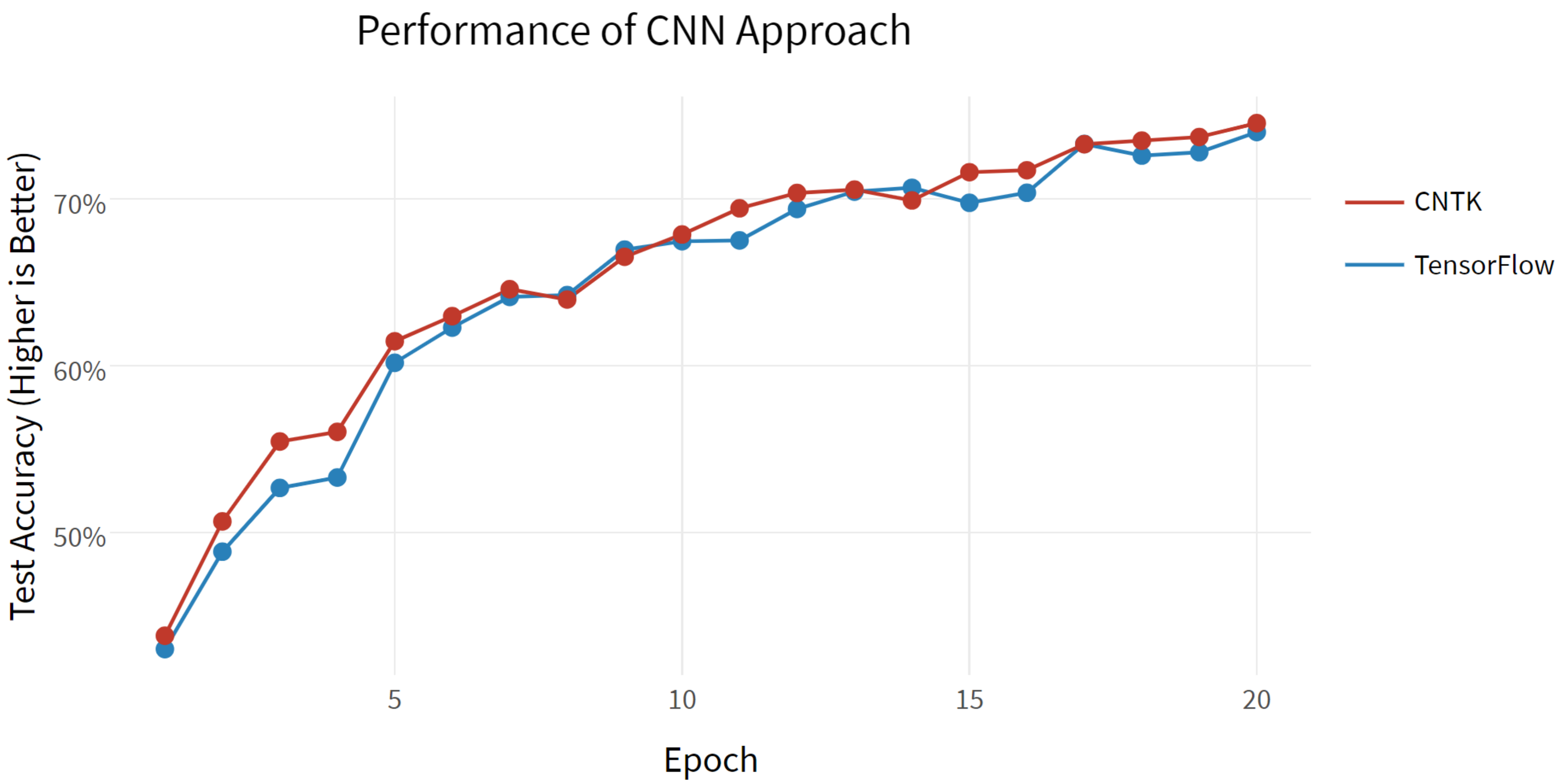
SPEED, FAST AND CHEAP TRAINING AND ACCURACY



Normally the accuracy *increases* as training proceeds; Bidirectional LSTMs take a long time to train to get improving results, but at the least both frameworks are equally performant.

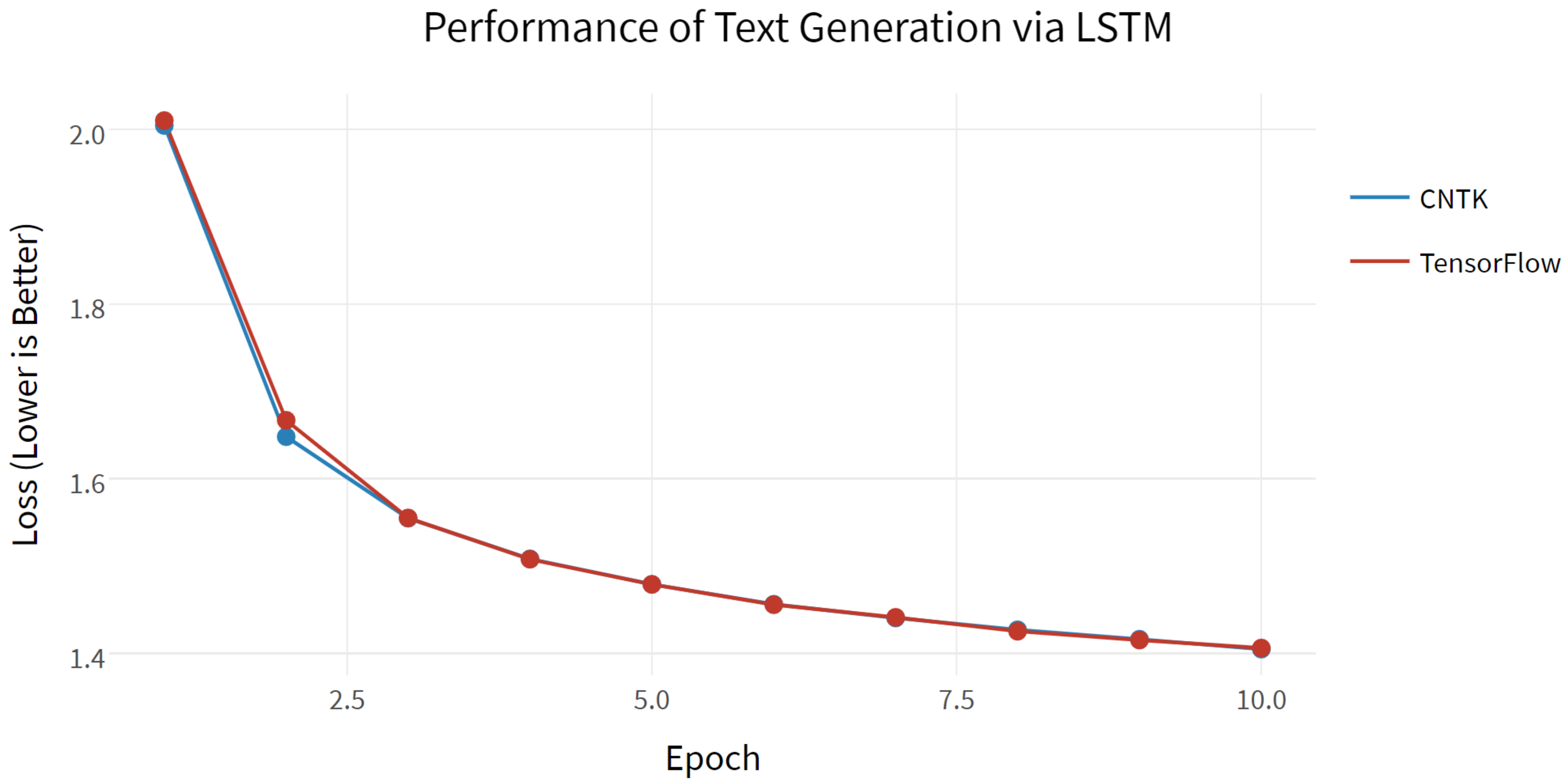
SPEED, FAST AND CHEAP TRAINING AND ACCURACY

ACCESS SOFTEK, INC



SPEED, FAST AND CHEAP TRAINING AND ACCURACY

ACCESS SOFTEK, INC



PLAN

Part I: how we choose CNTK + c#

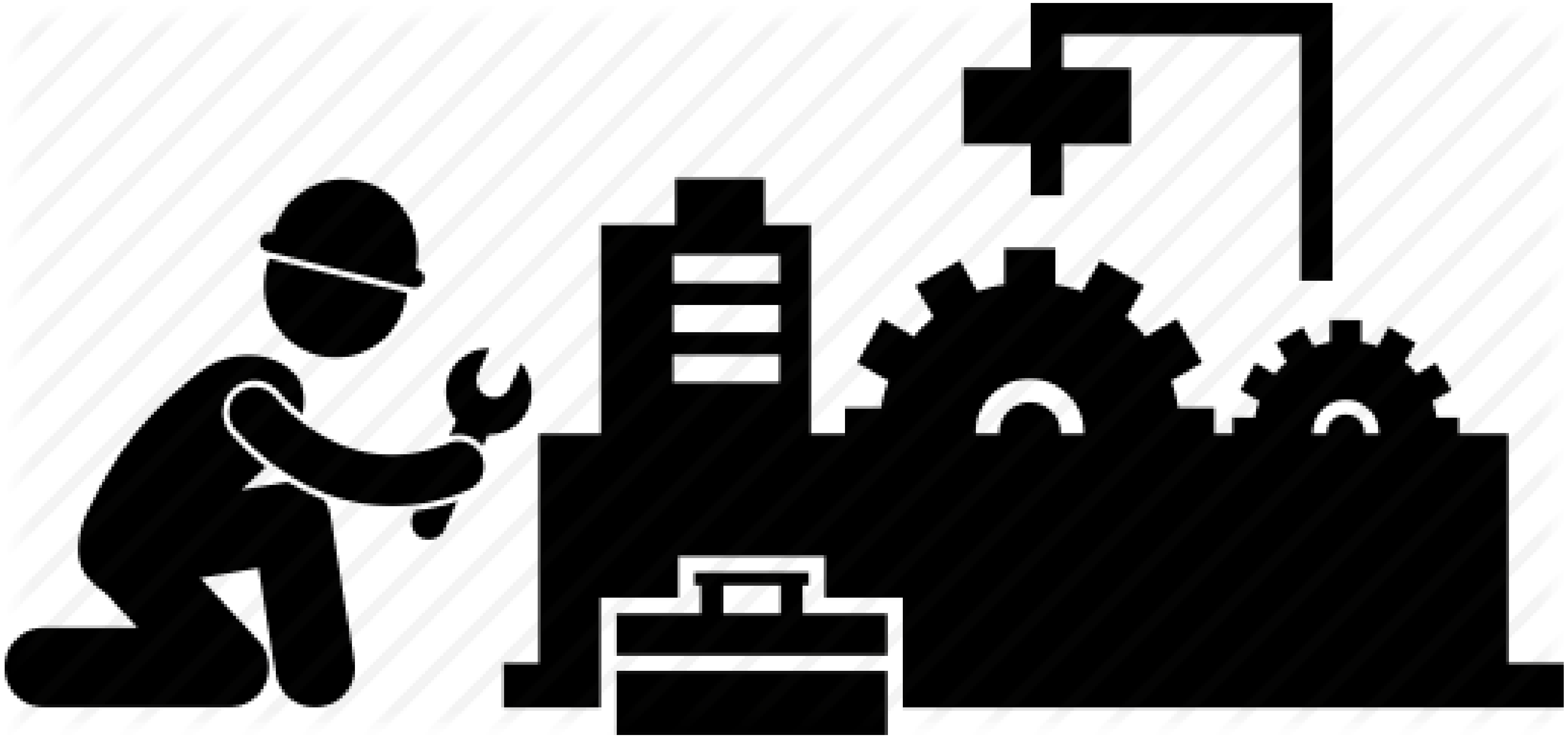
- Access Softek company, our tasks, data volumes
- 3 pillars of our requirements: speed, fast and cheap training and accuracy
- Comparison CNTK with other existing solutions

Part II: CNTK + c# in real life projects

- Fraud detection
- User Engagement
- Churn Prevention

PART II: CNTK + C# IN REAL LIFE PROJECTS

ACCESS SOFTEK, INC



PLAN

Part I: how we choose CNTK + c#

- Access Softek company, our tasks, data volumes
- 3 pillars of our requirements: speed, fast and cheap training and accuracy
- Comparison CNTK with other existing solutions

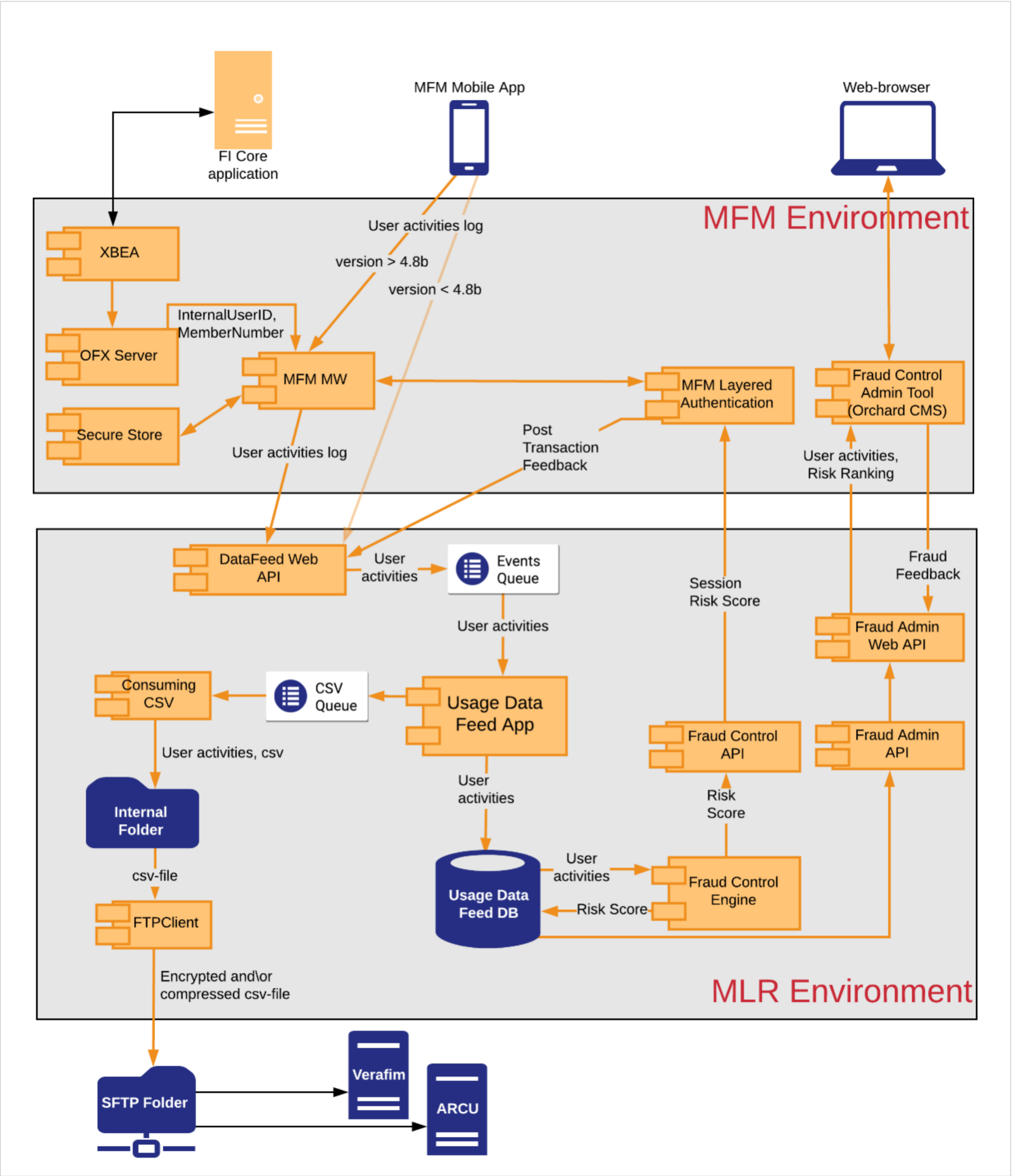
Part II: CNTK + c# in real life projects

- **Fraud detection**
- User Engagement
- Churn Prevention

PART II: CNTK + C# IN REAL LIFE PROJECTS: FRAUD DETECTION

ACCESS SOFTEK, INC

Architecture Overview



Overview

Detect, Predict, and Prevent Fraud, in real-time.

Fraud Control is designed to learn and react to each member's banking usage, to keep accounts safe and fraud under control.

•Detect suspicious behaviors

- Fraud Control's eyes are unsupervised and semi-supervised machine leaning techniques to cluster and classify out-of-pattern account-level behaviors

•Predict fraud risk

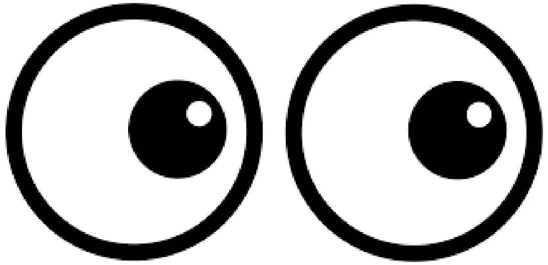
- Fraud Control's brain is neural network ensemble that learns to score suspicious behavior based on admin feedback to fraud discovery and user performance on event-triggered layered authentication

•Prescribe preventative countermeasures

- Fraud Control's hand is an API that integrates with the banking channel's layered authentication, to safeguard the app from fraudsters

•Discover new ways to control fraud

- Fraud Control's face is a reporting and visualization tool to help admins break down user activity, global alerts, specific fraud threats, and layered authentication performance.



What does Fraud Control see?

Real-time User Behaviors

- Member ID
- Session ID
- Date & Time
- User Operation
- Screen / Feature / Operation Group
- App Response
- Server Response Duration
- Transaction Value (where applicable)
- Application Version
- Client
- Device IP Address
- Device GPS (if enabled by end-user)
- Device OS Version
- Device Model

What does Fraud Control find?

- Surprising location
- Using VPN
- New or unconventional device
- Large or suspicious money movement
- Activity at atypical time or date
- Unusual user behaviors
- Suspicious app responses
- Etc...

What does Fraud Control do?

- Constantly learns from every data point sent from all devices and servers accessing the banking app

- Detects and scores suspicious behavior on the account

- Integrates with Layered Authentication to control access to app functions

- Constantly learns from feedback to improve its predictions over time

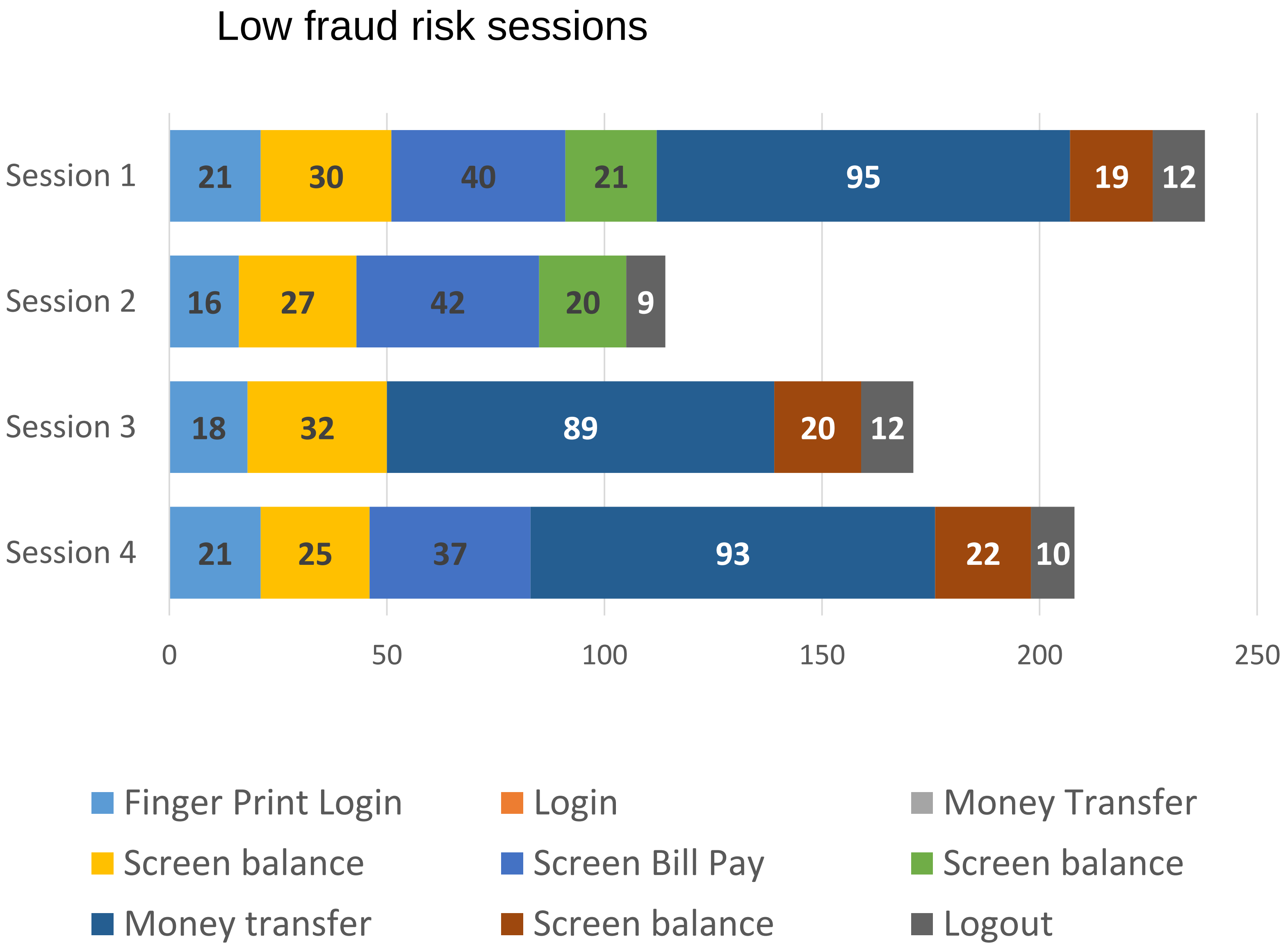
Five Fraud Risk Components

1. User Behavior Risk
2. Location Risk
3. Money Movement Risk
4. Time Risk
5. Device Risk

Concept is follows:

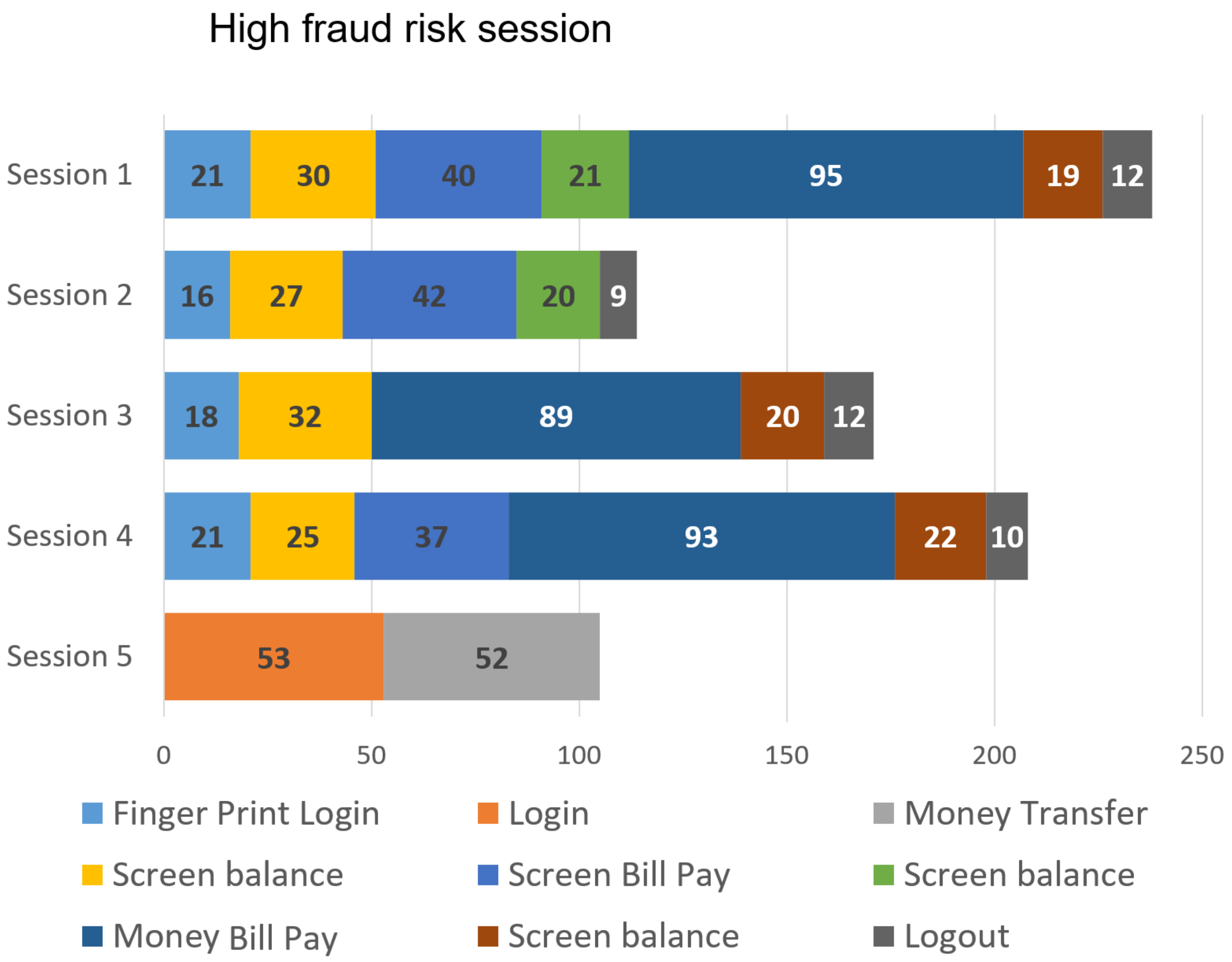
1. Separate 2 types of factors Linear and Matrix(ordered by steps)
2. For Linear - analyses distribution and “typical"
3. For Matrix - probability to meet event "A" on step "N"
4. Analyze by Model difference from current to typical
5. Aggregate all metrics on Session level
6. Choice how calculate 5 sub-scores in final Fraud Risk Score

User Behavior Risk



Risk Score
0.226
0.114
0.092
0.007

User Behavior Risk



Risk Score
0.226
0.114
0.092
0.007
0.981

Other Risk Components

Location Risk

- - lat-lon coordinates from device != lat-lon from IP-address
- - lat-lon changing faster than plane speed ~800km/hour

Money Movement Risk

- - Unusual amounts of money transfer
- - Unusual directions of money transfer

Device Risk

- - New device never used before
- - Unusual device combination, based on total device matrix

Time Risk

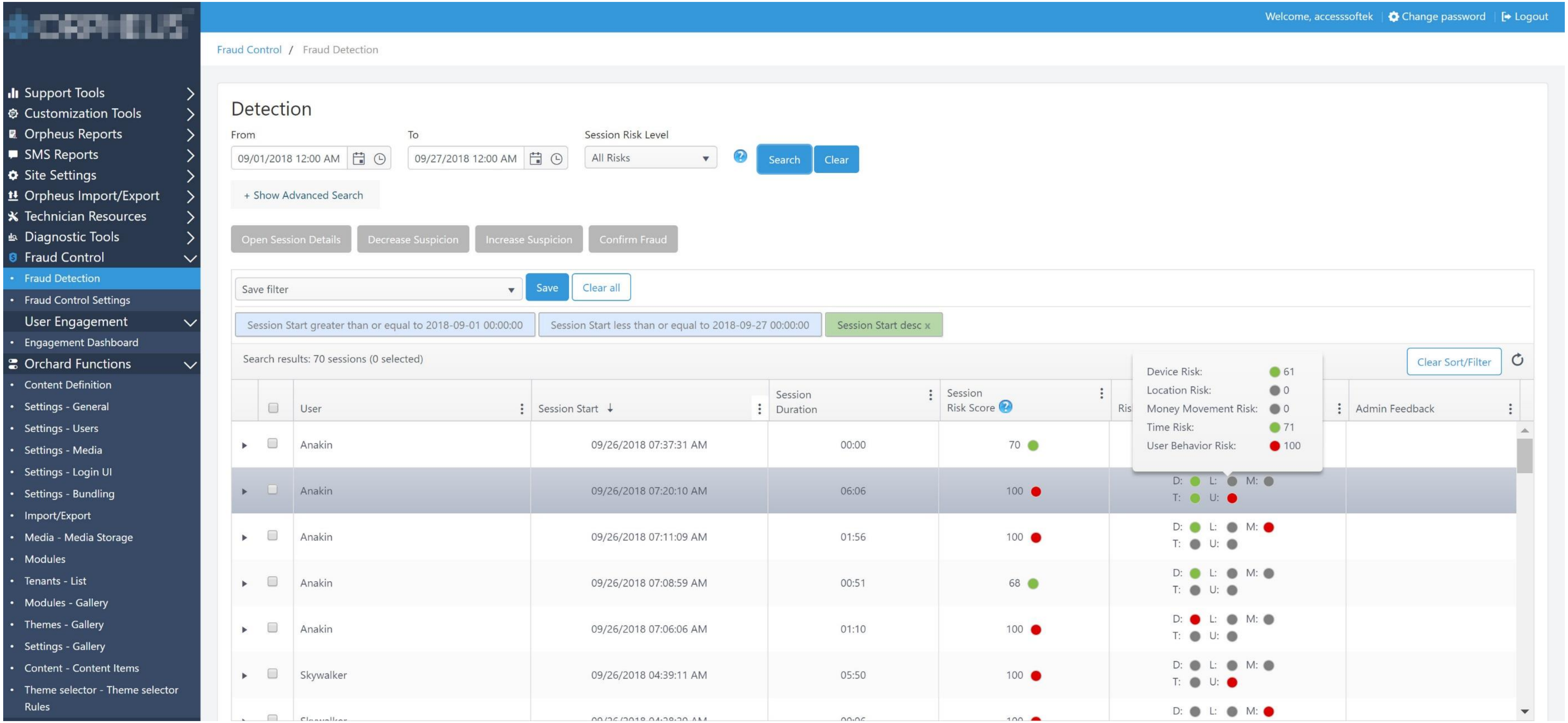
- - Unusual time of day for current customer
- - Unusual time of day for all customers

Strengths of the fraud detection tool

- ML Model for analyzing user behavior to detect out of pattern activity.
- Contains a system for learning from the feedback - if you return to model results marked as fraud or falsely identified fraud
- It analyses based on the individual user’s behavior
- Begins to work even on a small amount of data (the problem of a cold start is solved), does not require a large number of cases of confirmed fraud
- Done in real time on a session to stop fraud before it happens not after the fact historical review.
- We can monitor multiple channels, mobile, online and others.
- Configurable thresholds for Accept, Review, Stop transaction.

PART II: CNTK + C# IN REAL LIFE PROJECTS: FRAUD DETECTION

ACCESS SOFTEK, INC



PART II: CNTK + C# IN REAL LIFE PROJECTS: FRAUD DETECTION

ACCESS SOFTEK, INC

PLAN

Part I: how we choose CNTK + c#

- Access Softek company, our tasks, data volumes
- 3 pillars of our requirements: speed, fast and cheap training and accuracy
- Comparison CNTK with other existing solutions

Part II: CNTK + c# in real life projects

- Fraud detection
- **User Engagement**
- Churn Prevention

Engagement rate calculation

1) We have 37 features like session_interval, n_login_fails_login etc.
Based on the received features we calculate the values of 3 dimensions

$$\text{Frequency/recency} = \frac{1}{1+e^{-(\Sigma f_i * \log_2 f_i)}} \quad (1) \qquad \text{Depth or intensity} = \frac{1}{1+e^{-(\Sigma f_i * \log_2 f_i)}} \quad (2) \qquad \text{Money movement} = \frac{1}{1+e^{-(\Sigma f_i * \log_2 f_i)}} \quad (3)$$

2) When the values of Frequency-Recency, Depth-Of-Intensity, Money are calculated, we form the preTarget variable as Frequency-Recency + Depth-Of-Intensity + Money and sort the entire array of data by this value.
We take for training the model only 20% of the sample.

3) 10% of the top in our preTarget variable is denoted as the Target variable = 1,
10% of the bottom by the preTarget variable is denoted as the Target variable is 0.
This data is submitted to the input for model training.

4) For the forecast whole set of calculated features is submitted to the input of the model, and the Engagement Rate for the user is calculated for each user.

PART II: CNTK + C# IN REAL LIFE PROJECTS: USER ENGAGEMENT

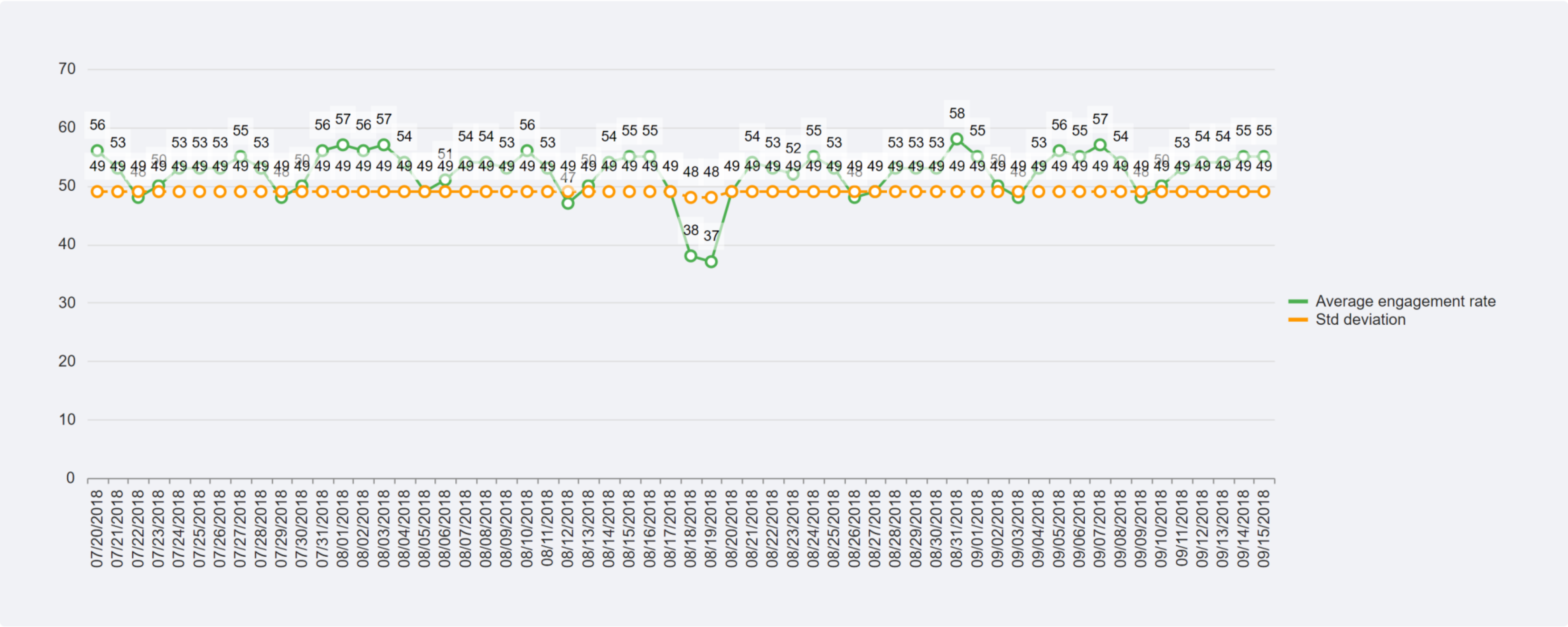
ACCESS SOFTEK, INC

- User Engagement
  - Engagement Dashboard
  - Churn Prediction Dashboard
- Data Analysis
  - Dashboards

Average Engagement Rate

Average and standard deviation of the Engagement Rate among all users

Engagement Rate description



PART II: CNTK + C# IN REAL LIFE PROJECTS: USER ENGAGEMENT

ACCESS SOFTEK, INC

User segmentation by Engagement

This chart displays segmentation of users based on their level of engagement with your app over the chosen time period

VIP	Users who use the application frequently (and recently) and use many features
Growing	Users who use the app periodically but only for some features
Newcomers	Users who use the app frequently but only for a few features
Lost	Users who previously used the many app features but rarely use the app lately
One-off	Users who previously used the app and didn't use many features

Engagement segments calculation

Users are classified into segments to a specific date using the following logic:

Frequency of use: This property indicates how frequently the user uses the app. It is calculated based on the following factors:

- average interval between sessions (for the previous [day, week, month])
- average session duration (for the previous [day, week, month])
- time since the user's last login (recency)

The possible values for each factor are 1, 2, or 3 where "3" represents high frequency and "1" represents low frequency.

Depth of usage: This property shows how deeply each user uses the app based on the number of features utilized in a session. It is calculated based on the following factors:

- number of unique events (for the previous [day, week, month])
- average time spent on one event (per [day, week, month] before the date)
- average, min and max event frequency - how often the event happens compared to other events (for the previous [day, week, month])
- variance of event frequency - deviation from the usual pattern of event frequency for the user (for the previous [day, week, month])
- number of devices used by the user (for the previous month)
- number of users per device (for the previous month)
- number of time the app was deleted and reinstalled (for the previous month)

The possible values for each factor are 1, 2, or 3 where "3" represents deep usage of the app and "1" represents superficial usage of the app.

PART II: CNTK + C# IN REAL LIFE PROJECTS: USER ENGAGEMENT

	Frequency of use		
Depth of use	13	23	33
	12	22	32
	11	21	31

(blue)	VIP
(dark green)	Growing
(yellow-green)	Newcomers
(light green)	Lost
(orange)	One-off

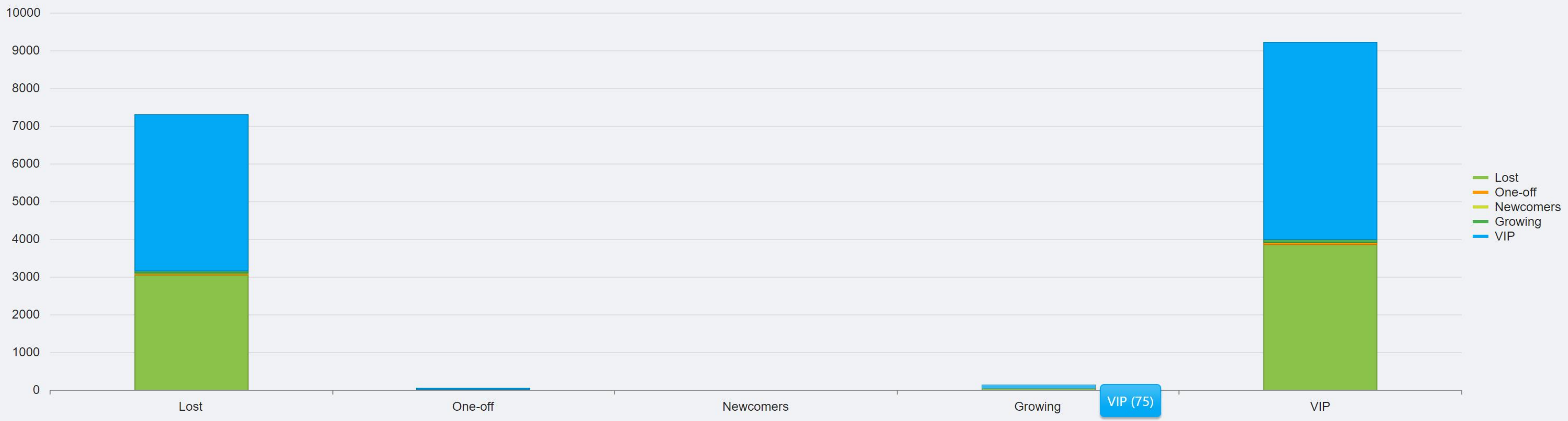
User segmentation by Engagement

This chart displays segmentation of users based on their level of engagement with your app over the chosen time period

VIP	Users who use the application frequently (and recently) and use many features
Growing	Users who use the app periodically but only for some features
Newcomers	Users who use the app frequently but only for a few features
Lost	Users who previously used the many app features but rarely use the app lately
One-off	Users who previously used the app and didn't use many features

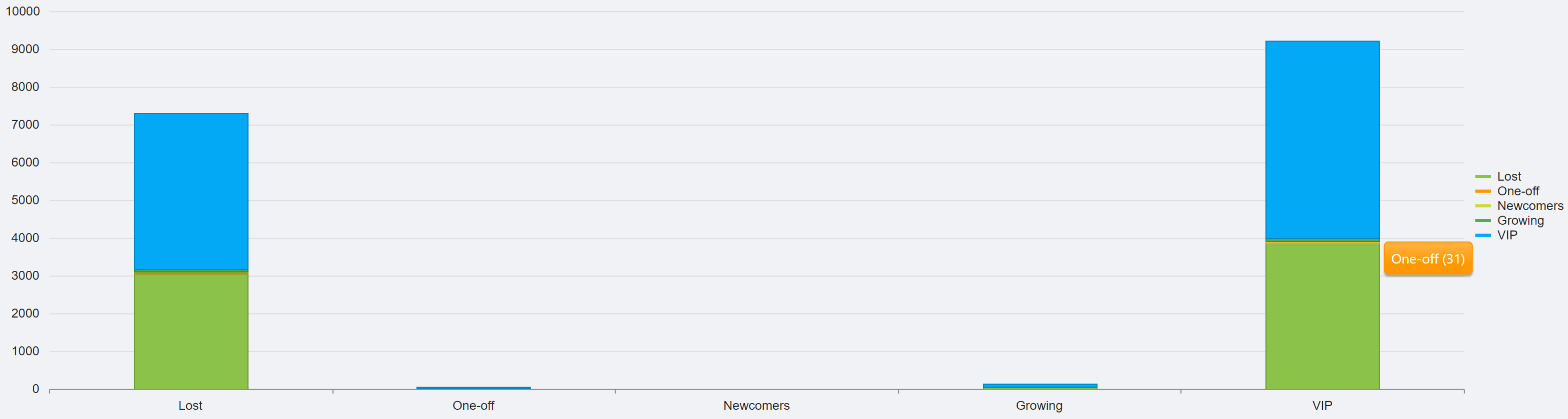
PART II: CNTK + C# IN REAL LIFE PROJECTS: USER ENGAGEMENT

ACCESS SOFTEK, INC



Migration 75 user(s) from segment "VIP" to segment "Growing"
Time since the user"s last login (recency) decreased on 121.18%
Min event frequency for the previous day decreased on 6.71%
Min event frequency for the previous week decreased on 6.71%
Min event frequency for the previous month decreased on 6.71%
Average event frequency for the previous day decreased on 6.05%

PART II: CNTK + C# IN REAL LIFE PROJECTS: USER ENGAGEMENT



Migration 31 user(s) from segment "One-off" to segment "VIP"
Number of unique events for the previous day increased on 12.83%
Number of unique events for the previous week increased on 12.83%
Number of unique events for the previous month increased on 12.83%

PLAN

Part I: how we choose CNTK + c#

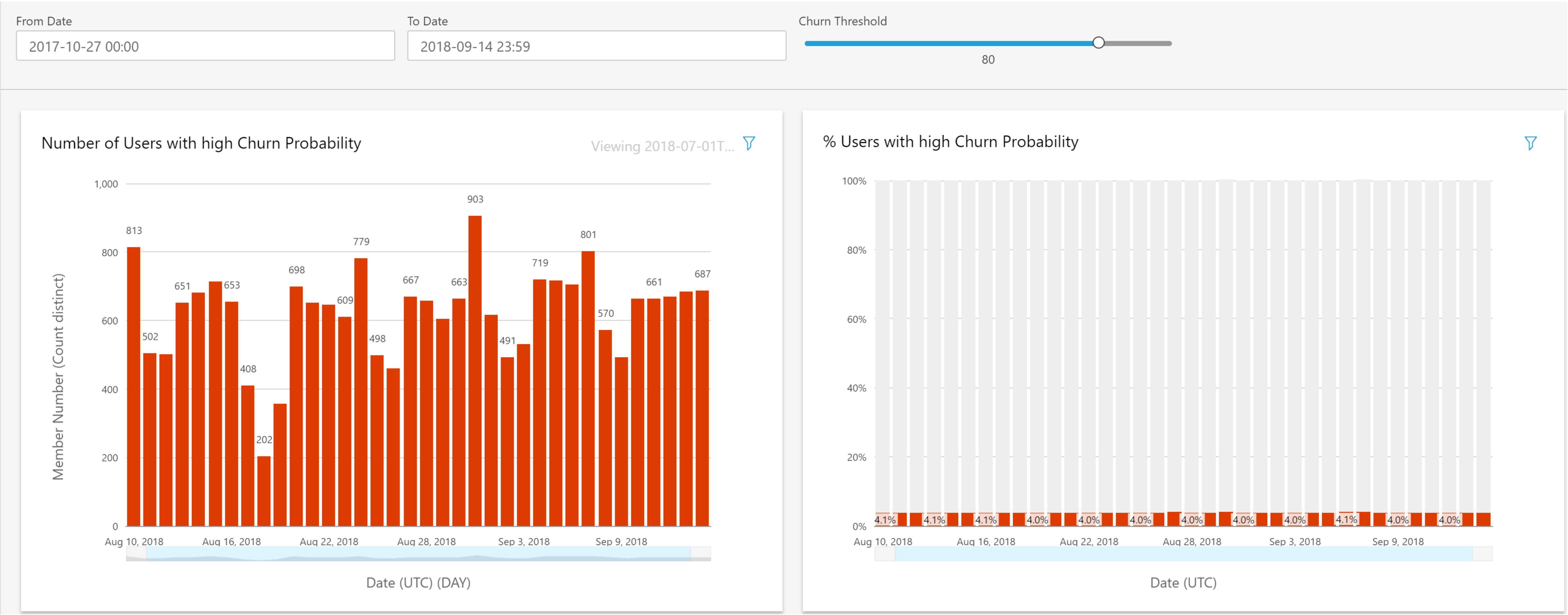
- Access Softek company, our tasks, data volumes
- 3 pillars of our requirements: speed, fast and cheap training and accuracy
- Comparison CNTK with other existing solutions

Part II: CNTK + c# in real life projects

- Fraud detection
- User Engagement
- **Churn Prevention**

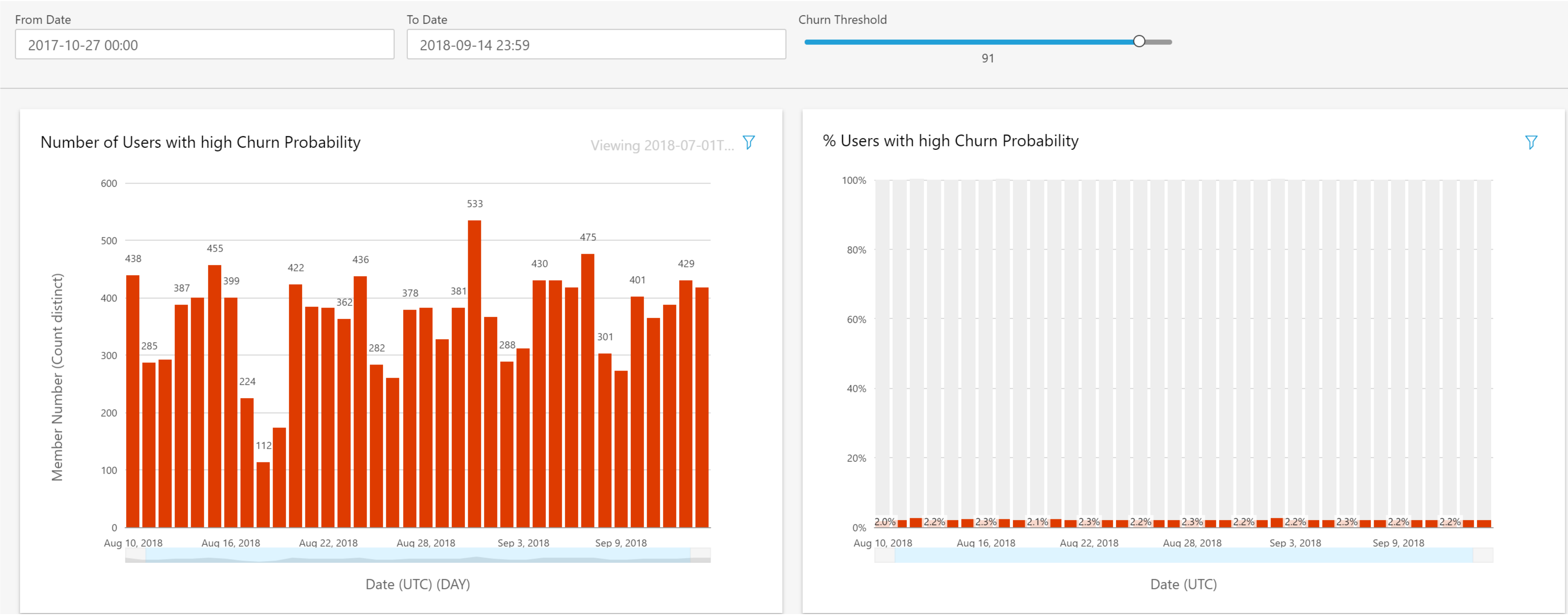
PART II: CNTK + C# IN REAL LIFE PROJECTS: CHURN PREVENTION

ACCESS SOFTEK, INC



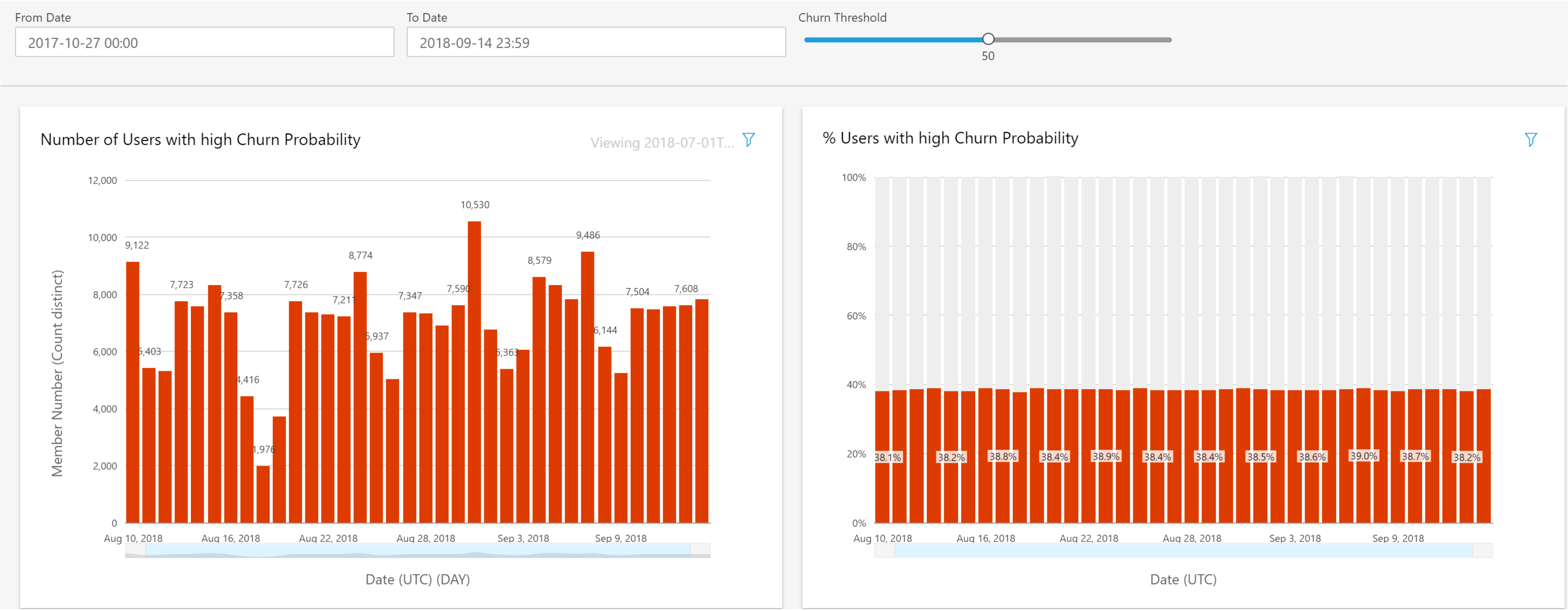
PART II: CNTK + C# IN REAL LIFE PROJECTS: CHURN PREVENTION

ACCESS SOFTEK, INC



PART II: CNTK + C# IN REAL LIFE PROJECTS: CHURN PREVENTION

ACCESS SOFTEK, INC



PART II: CNTK + C# IN REAL LIFE PROJECTS: CHURN PREVENTION

ACCESS SOFTEK, INC

Count of Member Number by Date and Probability Of Churn

Minimize

	Sep 14, 2018	Sep 13, 2018	Sep 12, 2018	Sep 11, 2018	Sep 10, 2018	Sep 9, 2018	Sep 8, 2018	Sep 7, 2018	Sep 6, 2018
Probability of Churn	Member Number	Member Number	Member Number	Member Number	Member Number	Member Number	Member Number	Member Number	Member Number
96	55	55	52	49	52	30	45	86	59
95	61	48	50	40	54	40	36	60	52
94	64	57	61	66	37	30	43	54	59
93	56	61	46	42	54	53	46	80	55
92	51	65	51	62	61	35	26	67	66
91	57	52	61	51	52	46	54	71	46
90	42	46	78	68	56	37	47	61	60
89	52	59	48	56	54	24	50	58	61
88	56	64	62	59	53	40	48	98	52
87	50	56	48	58	51	46	41	60	61
86	44	43	43	52	52	42	41	54	60
85	48	46	56	52	56	33	39	80	58
84	80	60	52	50	57	31	61	69	44
83	48	52	53	62	48	41	45	75	65
82									

QUESTIONS? COMMENTS? CONCERNS?

ACCESS SOFTEK, INC