Technical Disclosure Commons

Defensive Publications Series

September 13, 2018

VOICE CALL ANALYTICS PACKAGE FOR DETECTING FRAUDULENT ACTIVITIES AND ANOMALY DETECTION

Gyana Dash

Antonio Nucci

Aizhan Ibraimova

Vladimir Savostin

VOICE CALL ANALYTICS PACKAGE FOR DETECTING FRAUDULENT ACTIVITIES AND ANOMALY DETECTION

AUTHORS:
Gyana Dash
Antonio Nucci
Aizhan Ibraimova
Vladimir Savostin

ABSTRACT

Voice Call Anomaly Detection (VCAD) is described herein to detect inconsistencies in patterns. VCAD is an anomaly detection system which is based on a long short-term memory (LSTM) algorithm and statistical methods. By detecting inconsistencies in patterns, the models described herein may detect and alert user of unusual voice service behavior that if not properly corrected can degrade, and possibly disrupt, the voice service. The statistical and machine learning methods used by VCAD are generic and may be used for solving other time-series problems when using other type of logs such as call logs, game logs, application usage logs, etc. The VCAD proactive, predictive capabilities allow customers to either eliminate the issue altogether, or turn costly, unplanned outages into controlled maintenance windows.

DETAILED DESCRIPTION

Enterprises worldwide depend on a range of virtual conferencing and collaboration tools including audio and web conferencing, content sharing, and video conferencing, which is the highest impact form of collaboration. The strategic use of conferencing and collaboration yields numerous benefits including improved teaming and bonding with partners and peers, enhanced client retention, faster decision making, and more. The ever-increasing amount of telemetry and data (high definition audio and video), network complexity driven by the sky-rocking proliferation of voice and video applications, and heterogeneity of end-point systems makes the delivery of those services within the agreed level of quality and security a very complex problem in today's enterprises.

Though many solutions have been proposed for real-time monitoring and detecting abnormal behavior in voice communication networks, there are no acceptable solutions from an operational perspective. Anomaly detection has been widely studied in the context

of network data but only a few groups have published the application of their algorithms to voice call logs. In most cases, the anomaly detection problem is solved using static methodologies, which are methods that aim to detect differences between new data samples and historical averages. Those methods fall short in three key aspects.

First, they have the tendency to report too many false positives due to their tendency to confuse data noise with actual shifts in trending patterns. This translates into tools which are not prone for operational environments as too many reported incidents need to be further investigated manually by domain experts. Second, their main objective is to detect anomalies (i.e., a sudden change in behavior), rather than to predict the problem (i.e., detection of very early symptoms which will lead to the appearance of the problem in the future). If those symptoms could be detected early enough, it would give time for domain experts to properly analyze the patterns and, if action is needed, to implement the remediation to preclude the actual problem from occurring in the first place. Third, those methods ignore the challenges of real-time ingestion and analysis of noisy data.

In this regard, the algorithm needs to continuously learn the latest behavior of the service by ingesting in real-time new data, recalibrate the models by adjusting to legitimate dynamic shifts of behavior, and correctly assess whether any difference from those baselines have to be considered legitimate (i.e., service quality is still within an acceptable range) or abnormal (i.e., service quality will likely lead to a degradation or even worse the total disruption of the voice service at future times). Thus, in streaming, time-series data, anomalies give significant information in critical situations. Yet detecting anomalies in streaming data is a difficult task, requiring the model to process data in real-time, and learn while simultaneously making predictions.

To address the above challenges, a system is provided herein called Voice Call Anomaly Detection (VCAD). VCAD is an automated and unsupervised algorithm which ingests, cleanses and profiles voice call logs in real-time. It continuously learns the behavior of a voice service and accurately tracks its evolution. It then uses historical data and advanced machine learning algorithms to identify early symptoms leading to future problems. VCAD may also run in semi-supervised mode, in which domain experts can annotate events being investigated such that their actual feedback is automatically assimilated by VCAD learning models to improve both the accuracy and overall precision

over time. VCAD provides customers with the following business benefits: real-time anomaly detection, prevention of fraud attempts targeting the voice infrastructure, monitoring and prediction of voice resource utilization, and root cause analysis and troubleshooting.

The VCAD system may provide an end-to-end operational system which can detect anomalies and predict anomalies using Call Detail Records (CDR) and/or Call Management Records (CMR) data records as input. VCAD can be operated in a pure unsupervised mode, or semi-supervised mode, meaning that VCAD can leverage domain expert-provided feedback to recalibrate the models over time and hence improve its accuracy. Domain experts may annotate events after their investigation using the VCAD User Interface (UI) annotation module.

The preprocessing module uses a combination of time slicing and time wrapping to properly organize data samples into buckets which will improve the efficacy and accuracy of the statistical models for anomaly detection and anomaly prediction. Then it uses an Exponential Weighted Moving Average (EWMA) to score the importance of collected data records, providing more importance to the new samples, and less importance to the older samples. This way VCAD may track the legitimate dynamic shifts which are inherent in any voice service.

VCAD uses an extensive set of features, both numerical and quantitative, and operates multiple statistical models in parallel. Models may operate both on isolated features (univariate time series model) or across all extracted features (multi-variate time series models). Because numerical and quantitative features are different in nature, VCAD ay use two different modeling techniques: normal distribution fitting and residual analysis for numerical features, and long short-term memory (LSTM) for quantitative features.

VCAD uses robust data de-noising techniques to avoid triggering alerts in case of isolated divergence between observed and predicted values. VCAD uses a rolling time window (i.e., observation time window which moves forward over time and has several data samples contained within) to decide whether to trigger.

VCAD uses a Multi-Model Classifier to score every new monitored event. It consumes the output of all statistical models operating in parallel across all the monitored features to provide its final recommendation.

VCAD UI provides ANNOTATION which enables VCAD to be trained by domain experts as feedback is entered into the system, and hence extend the VCAD modalities of operation to semi-supervised. This may lead to a constant improving of accuracy and precision in both the identification of anomalies in real-time as well in the prediction of their future occurrence.

The VCAD methodology is very generic. In the remainder of this application VCAD is presented when using voice call logs collected from CDR and CMR systems, although the methodology may be generalized and applied to any other network and system logs which aim at addressing the aforementioned problems.

Figure 1 below illustrates the four major components comprising VCAD.

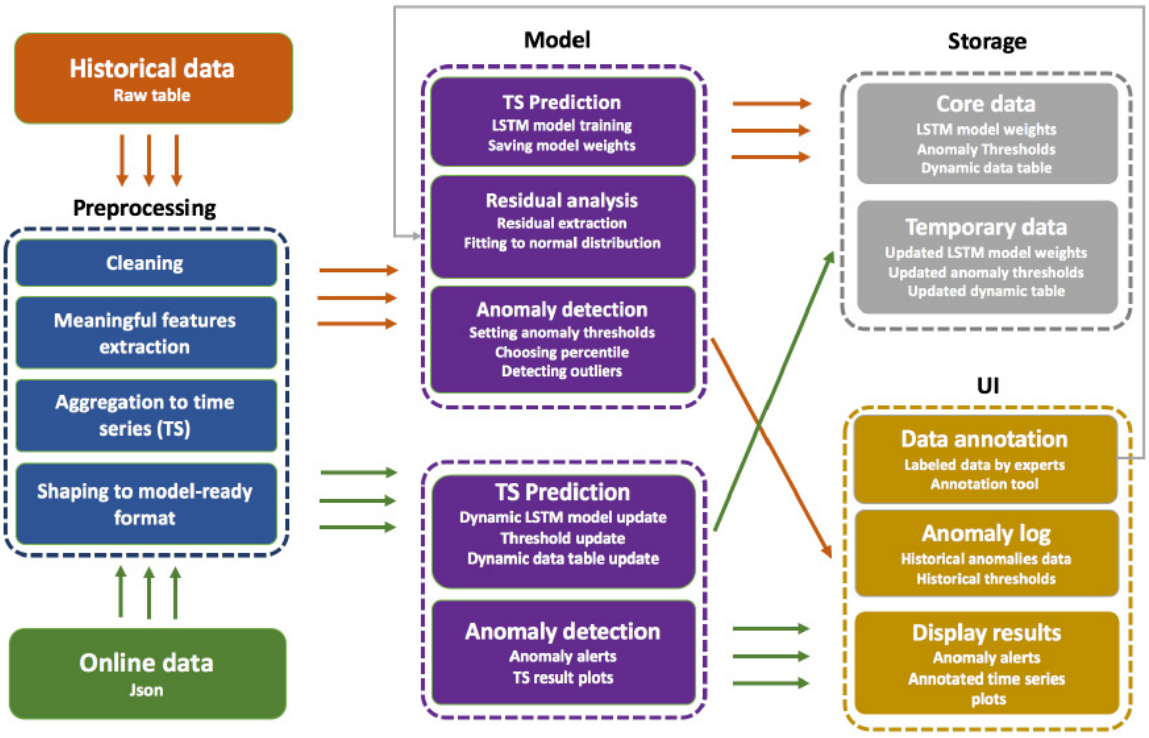

Figure 1: Methodology overview of the data flow from the raw data to the anomaly alerts displayed in the UI using VCAD method

The Preprocessing module collects voice call logs from CDRs and CMRs systems and then cleanses, formats, and aggregates them based on time. In voice communication networks, the statistics of voice calls are very time-sensitive, and every {day, time-bin} may have a behavior which is way different from the behavior of other time-bins. For example, call duration and call volumes are different during weekdays and weekends, and between business hours and night hours. Moreover, volumes of calls in enterprises are

much lower during holiday times than other days, and often spike during a new product release or thereafter for support.

The Preprocessing module slices the time into temporal bins (e.g., {day, time bins}) (time-slicing), and then aggregates data samples of the same time bin across multiple and same days (time-wrapping). For example, it can slice time in weekdays (Monday through Sunday) and each day into one-hour time bins (e.g., 8-9 am, 9-10 am, …, 7-8am). It then uses a rolling window to aggregate all samples for the same day and one-hour together. In this example, if for every "Monday 8-9 am" VCAD collects N data samples and it uses a four-week rolling window, the Preprocessing module may consider a total of 4xN data samples to generate the statistical model capturing the dynamics of Monday 8-9 am.

Because the newest observations are more relevant (more attaining to present behavior) than past observations, the Preprocessing module uses the EWMA technique operated by two attention coefficients to weight the data samples during the aggregation phase: w1 and w2, 0<= w1<=1, 0<= w2<=1 and w1 + w2 = 1. In the previous example, when aggregating the 4xN samples to generate the model for {Monday, 8-9 am}, it will multiply older samples with lower weights and newer samples with higher weights. This way, the model keeps refreshing with time and hence is able to track shifts in the dynamics of the voice service.

The processed data is then used as a table of standard values and is used by the Model component to properly predict future anomalies. This module extracts both numerical and categorical features which will be used by the Model generation component. The Model component primarily generates predictive models and performs analysis on the data to identify anomalies (comparison with the table of standard values created and constantly refreshed by the Preprocessing component). The Model component uses two different types of models for the two types of features extracted. For numerical features it uses a combination of normal distribution fitting and residual analysis, while for quantitative features it uses LSTM.

VCAD does not model generate models using only each feature in isolation, but also generates models which consider all features available from the features pool. Hence, VCAD is both a univariate time series and multivariate time series anomaly detection

system. The following example is provided to help illustrate the importance of the multivariate models.

Two quantitative features may be extracted: (i) number of calls, and (ii) calls duration. There may be cases where even though the distributions of the two features is normal (e.g., follow their historical univariate distribution), the join distribution may not, (e.g., a sudden spike in number of calls with a call duration of X seconds). Events of this type cannot be detected by univariate models but may be detected by VCAD using the multivariate models. The models used by VCAD may be easily generalized to incorporate other telemetry types and features as available such as voice recordings (audio files) and syslogs (device / system logs) to expand the number of problems it may detect and even better localize the problem in the network.

The Storage component is primarily in charge of persisting internal parameters of the generated models, such as their weights, thresholds, attention coefficients, dynamic data tables, etc. Finally, the User Interface (UI) component exposes the detected anomalies and related logs to domain experts who can review, label and annotate the anomaly reported. This is used by VCAD to constantly execute model auto-calibration and model auto-refreshing. Because of this closed feedback loop between domain experts and the system, VCAD may be trained in real-time and hence leverage unique feedback from domain experts. In this mode of operation, VCAD may retain the correct properties of its models to keep triggering true anomalies (as labeled buy domain experts) while recalibrating other properties to avoid triggering on false positives. This is done automatically and continuously as new feedback is entered into the system. This approach improves the quality of the prediction over time as the system absorbs new feedback provided by domain experts.

Figure 2 below illustrates five analytical modules comprising the VCAD system: VCAD Data Modeling, VCAD Time Series Prediction, VCAD Domain Voice Record Analysis, VCAD Anomaly Analysis and VCAD Anomaly Detection.

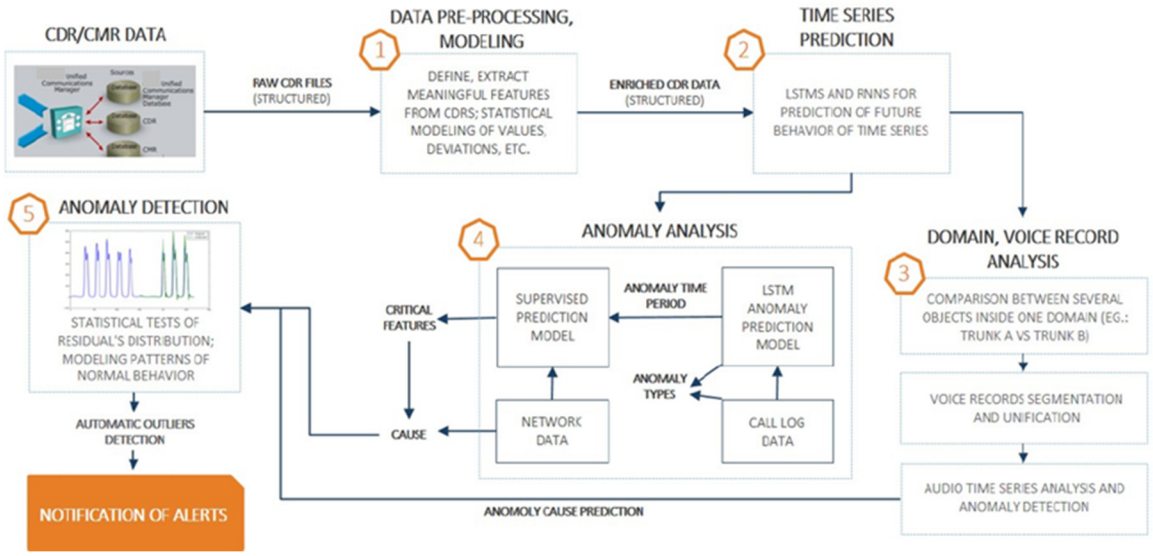

Figure 2: VCAD Information Flow Diagram illustrating five foundational components including Data Modeling, Time Series Prediction, Domain Voice Record Analysis, Anomaly Analysis, and Anomaly Detection

The VCAD Data Processing and Modeling receives as an input the data that needs to be processed in order to detect anomalies. In one example, call logs may be used (e.g., CDRs and CMRs).

Figure 3 below illustrates an example of processed CDRs (e.g., important fields extracted from a raw CDR record which has a flat format).

origLegCallIdentifier	Identifier of Incoming Call Leg (needed to identify specific call, not required for model itself)
dateTimeOrigination	Datetime when call came to the system in unixtime format
origIpAddr	Signalling IP Address of Calling party (Incoming Call Leg)
origCause_value	Disconnect Cause for Incoming Call Leg
origMediaTransportAddress_IP	Media IP Address of Calling party (Incoming Call Leg)
origMediaTransportAddress_Port	Media UDP Port of Calling party (Incoming Call Leg)
destLegIdentifier	Identifier of Outgoing Call Leg (needed to identify specific call, not required for model itself)
destIpAddr	Signalling IP Address of Called party (Outgoing Call Leg)
finalCalledPartyNumber	Called Number - required for Tall Fraud model (specific prefix identifies international or other interesting destinations)
destCause_value	Disconnect Cause for Outgoing Call Leg
origMediaTransportAddress_IP	Media IP Address of Called party (Outgoing Call Leg)
origMediaTransportAddress_Port	Media UDP Port of Calling party (Outgoing Call Leg)
dateTimeConnect	Datetime when call was connected in unixtime format
dateTimeDisconnect	Datetime when call was disconnected in unixtime format
duration	Call Duration in seconds
origDeviceName	Name of Calling Device as it's confgured by administrator in the phone system
destDeviceName	Name of Calling Device as it's confgured by administrator in the phone system

Figure 3: The important column names for CDR

Every time the Communication Manager is engaged in an active call (e.g., receives or starts a new call), it generates a CMR and CDR record as soon as the call is terminated. CMR records contain quality of service (QoS) and diagnostic information about the call

and associated statistics such as data sent and received, jitter, latency, and lost packets during the call. Conversely, CDR records contain information such as call origination, call destination, the date and time the call was started, the time it actually connected, and the time it ended. The VCAD system uses both CMR and CDR records to extract engineering features which may be used by the Model component to derive the behavioral predictive models.

Figure 4 below illustrates an example of a quantitative feature behavior over time (number of calls) for a user.

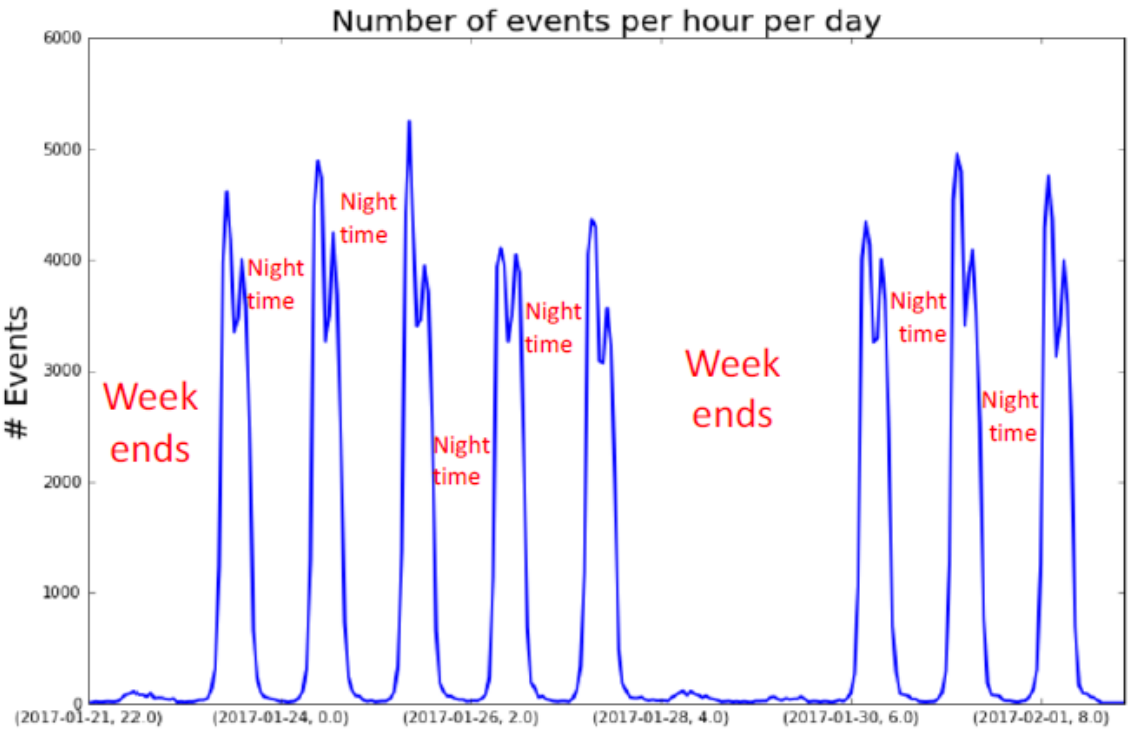

Figure 4: The number of events (calls) happened per hour per day for a user; the time diapason with the best patterns is chosen.

By working closely with domain experts, two types of engineering features may be extracted: numerical and quantitative. The numerical features capture specific numerical values about each call. Examples of numerical features are (i) call duration, (ii) waiting time, (iii) time between calls, etc. The quantitative features capture call activities at a higher level by aggregating statistics across calls and time bins (week, day, hour and minute time bins). Examples of quantitative features are (i) active call minutes in each trunk, (ii) number of calls, (iii) number of calls with bad termination cause codes, (iv-v) number of long and short calls, etc. The number of calls in the user is clearly different when comparing

weekends and weekdays or business hours and closed-business hours (i.e., night hours). Once the data is pre-processed and the features above are extracted by the Preprocessing module, the Model component automatically generates models which capture the normal historical behaviors. This is then used to predict future states and alert of any divergence from them (i.e., outliers).

As described in the previous section, any voice call associated with a CDR/CMR record which contains some features that are expressed / represented by their numerical values. Those key:value pairs, where the corresponding value is of a numerical form, are used by VCAD as numerical features. For those features, VCAD may use residual analysis to detect behavioral anomalies. Residuals are estimates of experimental errors obtained by subtracting the observed values from the predicted values. The observed values of each numerical feature are stored in the tables of standard values which are computed and regularly refreshed by the Processing module. The predicted values are extracted from a distribution which best fit the empirical data. In VCAD the normal distribution is used as the distribution which best fits the observed data. The normal distribution was the end result of a thorough sets of experiments where the observed data for all the numerical features extracted by VCAD was tested using the standard Kolmogorov-Smirnov distribution fitting test. In short, the Kolmogorov-Smirnov uses a null hypothesis H_0, which states that the observed data is normally distributed, and an alternative hypothesis H_1, which states that the data is not normally distributed to score the quality of the distribution fitting. More specifically, this test checks if any given sample is extracted from a population with a specific distribution using the function shown in Equation 1:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{[-\infty,x]}(X_i), \quad \text{Equation:1}$$

where $I_{[-\infty,x]}(X_i)$ is the indicator function. If $X_i \leq x$, then the indicator function is equal to 1 and equal to 0 otherwise. The Kolmogorov-Smirnov test on a given cumulative distribution function F(x) is defined as in Equation 2:

$$S = \sup_x |F(x) - F_n(x)|, \quad \text{Equation: 2}$$

where $\sup_x$ is the operator of supremum. For all the numeric features, it is confirmed that the null hypothesis H_0 was not rejected based on the obtained p-value and

predetermined significance level. Hence, it can be safely concluded that the empirical histogram (i.e., histogram of observed data records), is indeed well approximated by the Normal distribution.

As an example of the quality of the distribution fitting, Figure 5 below illustrates both the empirical distribution (blue vertical bars) and the normal distribution which best fit the data (red dotted line) for feature of the number of calls for a trunk in a user.

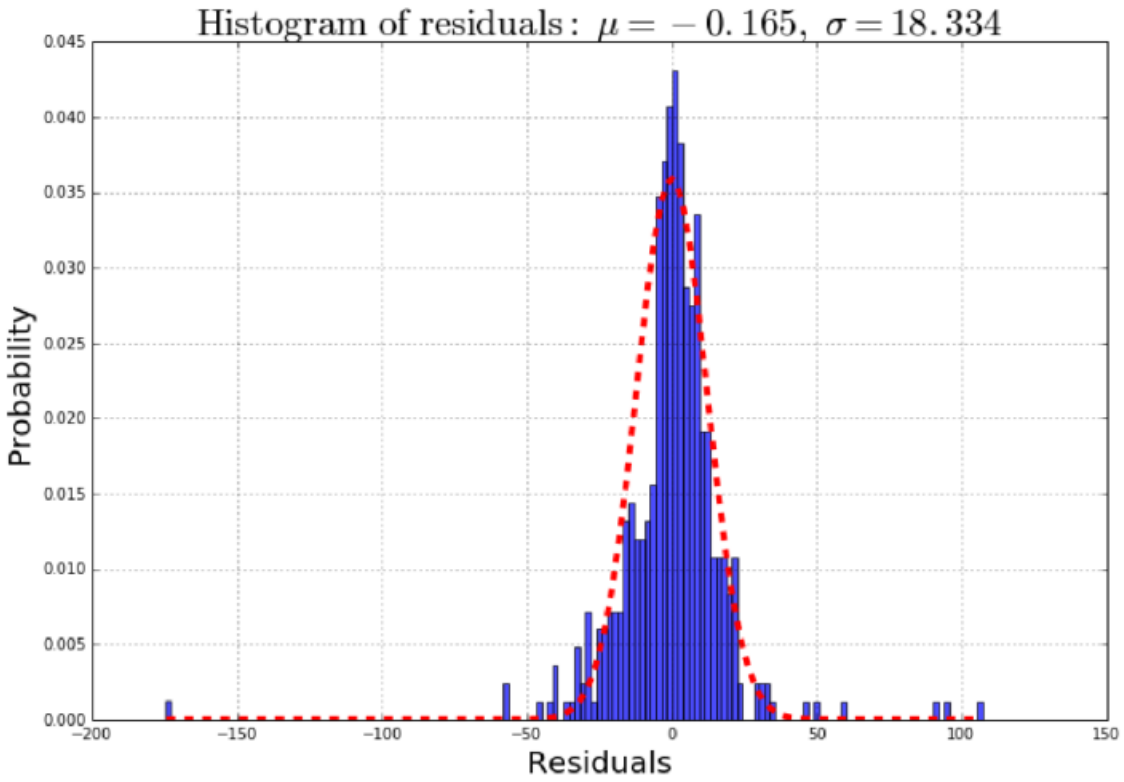

Figure 5: The histogram of the residuals between actual (blue vertical bars) and predicted (red dotted line) values for the number of calls in one trunk of a user.

The best fitted normal distribution (the optimal normal distribution) has a mean equal to -0.165 and a standard deviation equal to 18.334. The closer the data samples are to the mean value of the optimal normal distribution, the more they are labeled as normal. Conversely, the further away the data samples are from the mean (and hence closer to the tails of the normal distribution) the higher will be their suspiciousness score. For every value bin (the x-axis is organized into equally separated bins) the residual fitting model also learns the distribution of distances between the empirical histogram and the fitted normal distribution (i.e., residuals). This is then used to auto generate bin-specific thresholds to trigger alerts. The VCAD user may also customize the auto-learned thresholds

as an input parameter to drive VCAD to generate more (lower the thresholds with a down scaling factor) or fewer alerts (increase the threshold value by a scaling up factor).

Figure 6 below illustrates an example of LSTM in action on the number of calls in a user. the number of calls with BAD termination in a Customer B (Figure 7) and the number of calls with BAD termination codes in a Customer B for different time diapasons (Figure 8).

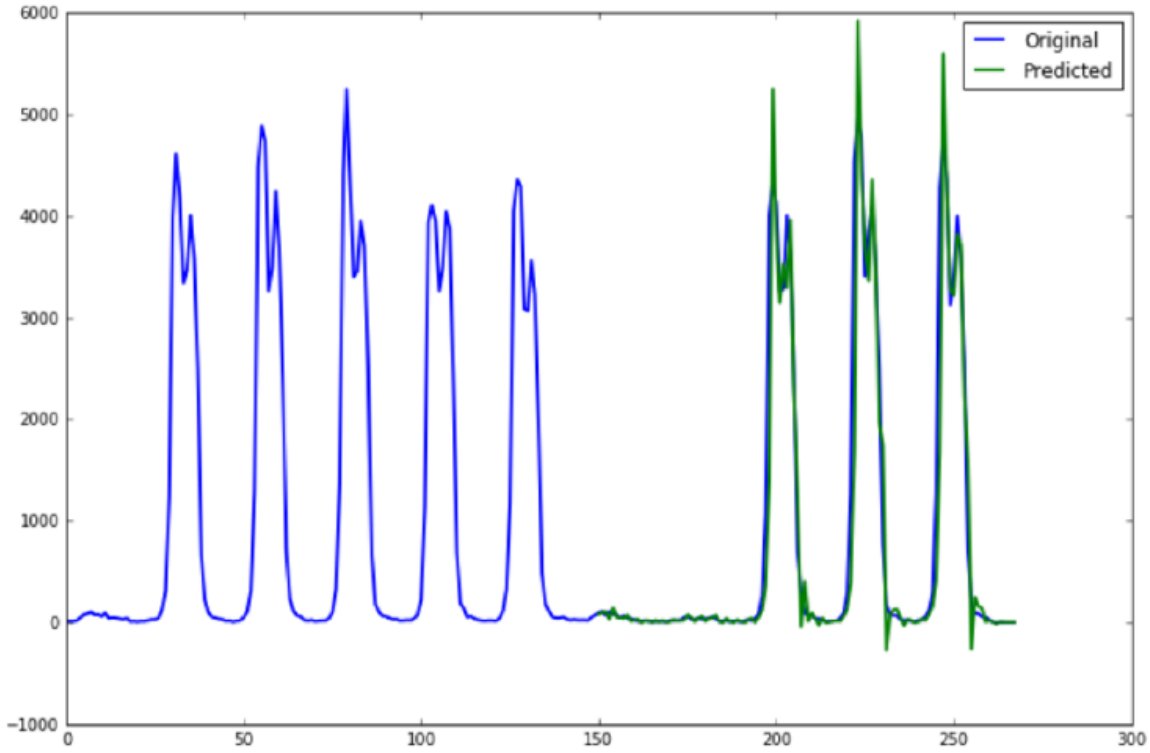

Figure 6: Temporal seasonality of number of calls for a user. The LSTM model is trained on the first five business days (blue line) and predicts very accurately (green line) the number of calls for the three business days of the following week.

Next, provided is a method to compute the number of existing outliers when comparing the empirical histogram and the compute optimal normal distribution for any given numeric feature. To achieve this, the Generalized Extreme Studentized Deviation (ESD) test is used. The upper bound for the number of outliers k is a parameter that is preconfigured in advance. Therefore, the null hypothesis that the data has no outliers is tested against the alternative hypothesis that there are at most *k* outliers. The test statistics is shown in Equation 3:

$$S_i = \frac{\max_i |x - x_i|}{\sigma}, \quad \text{Equation: 3}$$

where x and σ are the mean and the standard deviation of a sample correspondingly. After one iteration, the i-th observation which maximizes $|x-x_i|$ is removed and the corresponding statistics are recomputed. The same process is repeated k times. For each of the statistics S_i, i = 1, 2, …, k, the critical values λ_i. are calculated as below:

$$\lambda_i = \frac{(n-i)t_{p,n-i-1}}{\sqrt{(n-i-1+t^2_{p,n-i-1})(n-i+1)}}, \quad \text{Equation: 4}$$

where

$$p = 1 - \frac{\alpha}{2(n-i+1)}, \quad \text{Equation: 5}$$

and α is the user specified significance level, $t_{p,n-i-1}$ is the 100 ∗ p percentage of the t-distribution with n-i-1 degrees of freedom. The largest i which satisfies the condition $S_i > \lambda_i$ corresponds to the number of outliers that this model will report.

Figure 7 below illustrates an example of LSTM in action on the number of calls with BAD termination in a user. and the number of calls with BAD termination codes in a Customer B for different time diapasons (Figure 8).

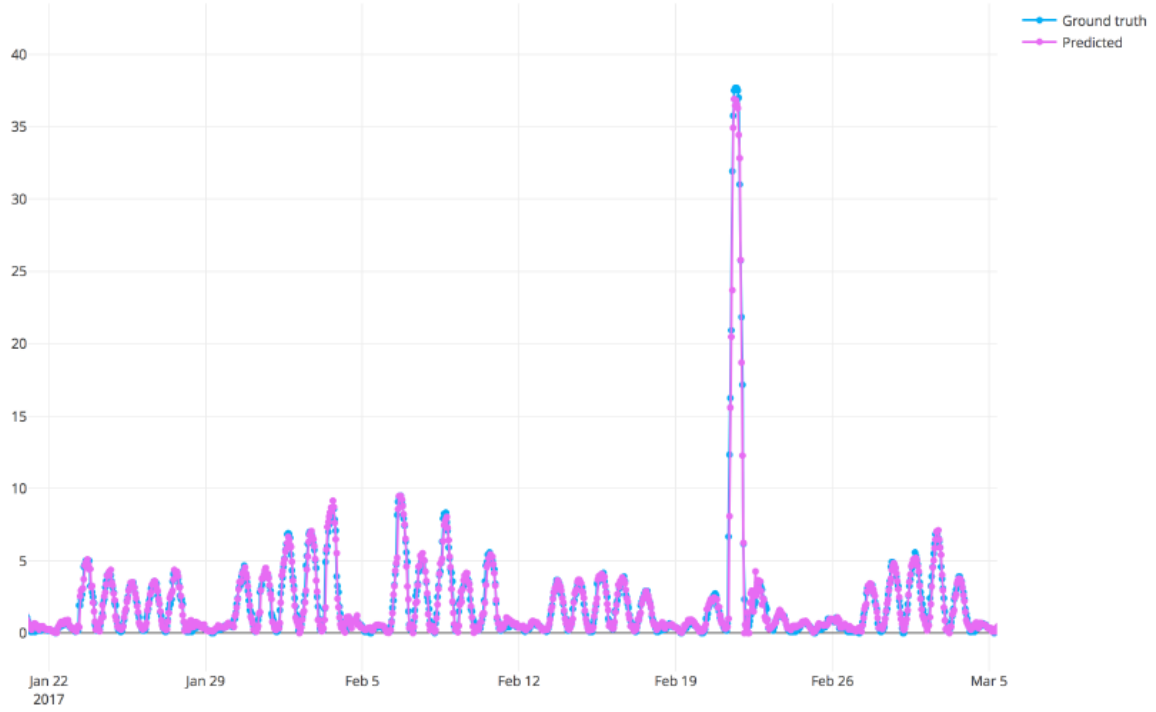

Figure 7: The number of calls with bad termination codes in one of the clusters for a user for different time diapasons. The blue line corresponds to the original values and the purple line corresponds to the predicted ones.

The second family of features is the quantitative features. For learning and predicting the behavior of quantitative features, VCAD uses a deep learning algorithm known as LSTM. For those features, the VCAD Model component applies LSTM on the preprocessed data which may extract hidden trends by auto-selecting the features and auto-generating the associated model for future predictions. The LSTM algorithm is a powerful algorithm which is designed to remember the important information from the past (which features to use) and forget all unnecessary information (which features to drop) while keeping in memory both the long and short temporal dependencies among the features.

In the above cases, an aggregation time bin of one-hour for one user and a bin of five minutes for the other user are applied. The LSTM does a great job in learning from empirical observations the temporal feature trends and uses the discovered trends to accurately predict their future values. In order for the Model component to generate predictive models which are accurate over time, it is important that it uses the latest empirical observations. VCAD uses the Preprocessing module to compute this. The Preprocessing module use an EMWA to weight more recent observations (standard values in the table of standard values maintained by the Preprocessing component) and weight less the older observations. It does this at regular times t as captured by Equation 6:

$$v_t(w_1, w_2) = w_1 * v_{t-1} + w_2 * a_t \qquad \text{Equation: 6}$$

In Equation 6, the $v_t(w_1,w_2)$ represents the updated standard value for the time t, v_{t-1} represents the standard value for the previous time t-1, a_t is the new ground truth value for time t, and w_1, w_2 are the corresponding attention coefficients used to weight the importance of the samples corresponding to the two times t and t-1, i.e., $0 \leq w_1 \leq 1$, $0 \leq w_2 \leq 1$, and $w_1 + w_2 = 1$.

Figure 8 below illustrates an example of LSTM in action on the number of calls with BAD termination codes in a user for different time diapasons.

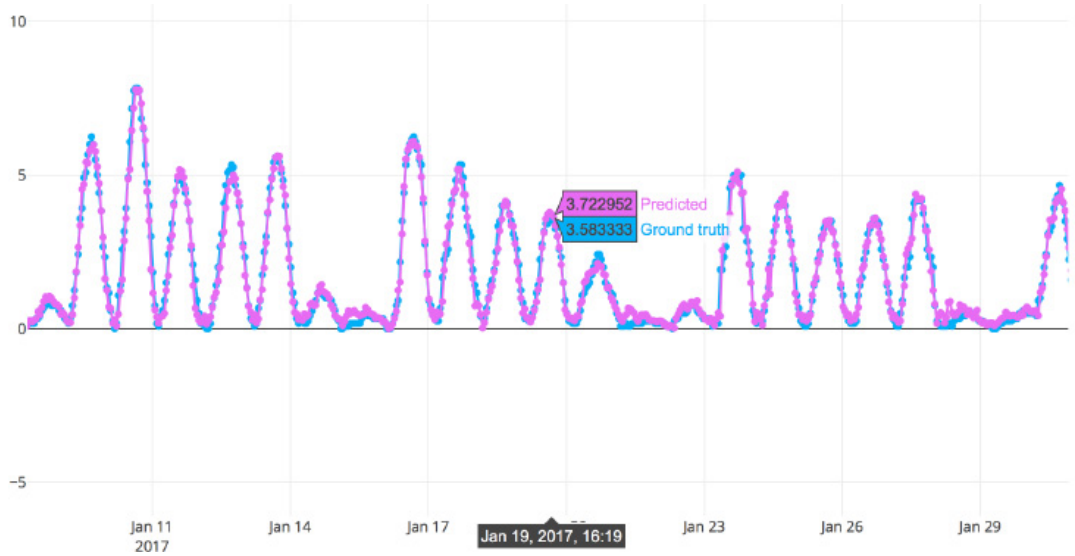

Figure 8: The number of calls with bad termination codes in one of the clusters for a user for different time diapasons. The blue line corresponds to the original values and the purple line corresponds to the predicted ones.

In order to find the optimal coefficients w_1 and w_2 for Equation 6, the loss function described in Equation 7 is optimized and defined:

$$\mathcal{L} = \min_{w_1,w_2} \mathrm{E}_{t\sim T}(a_{t+1} - v_t(w_1, w_2))^2 \qquad \text{Equation: 7}$$

where data points t are sampled from the historical data T and E is the operator of the expected value.

Now, let d_t be the difference between the predicted value for the moment t (i.e., f_t),and the standard value of the previous time combination v_{t-1}:

$$d_t = f_t - v_{t-1} \qquad \text{Equation: 8}$$

It is intuitively clear that if the value of d_t is high, then the difference between the predicted estimate and the previous standard value is high, hence abnormal. As a result, it is important to find the "right" thresholds not to consider too many values as abnormal.

The historical data may be leveraged for that purpose. The values of the points before the moment t may be used to predict the value f_t for the same time moment t. The actual value a_t for the moment t may be used to compute the anomaly score s_t shown in Equation 9 for any time moment before time t as:

$$s_t(a_t, f_t) = \frac{2 * |a_t - f_t|}{|a_t| + |f_t| + \varepsilon} \qquad \text{Equation: 9}$$

Equation 9 may be used to quantify the goodness of the prediction when compared to observed data samples for the exact same time t. In Equation 9, ε is a small correction factor (it is a constant, $0 < \varepsilon < 10^{-p}$, where $p \epsilon N$. The bigger p is, the better it is for the score value, so that $\varepsilon \sim 0$). This correction ensures that $s_t(a_t,f_t)$ always exists (i.e., if both $a_t = 0$ and $f_t = 0$ the anomaly score may lead to numerical rounding problems without the correction factor).

The range of the anomaly score s_t varies from 0 to 1, where $s_t = 0$ means a perfect match between the predicted values and the empirical observation (i.e. $a_t = f_t$). Conversely, $s_t \rightarrow 1$ is symptomatic of a sharp divergence between the predicted and observed value which implies the data sample is an outlier.

Figure 9 below illustrates the derived error rate of the prediction for the same feature and user used for Figure 8.

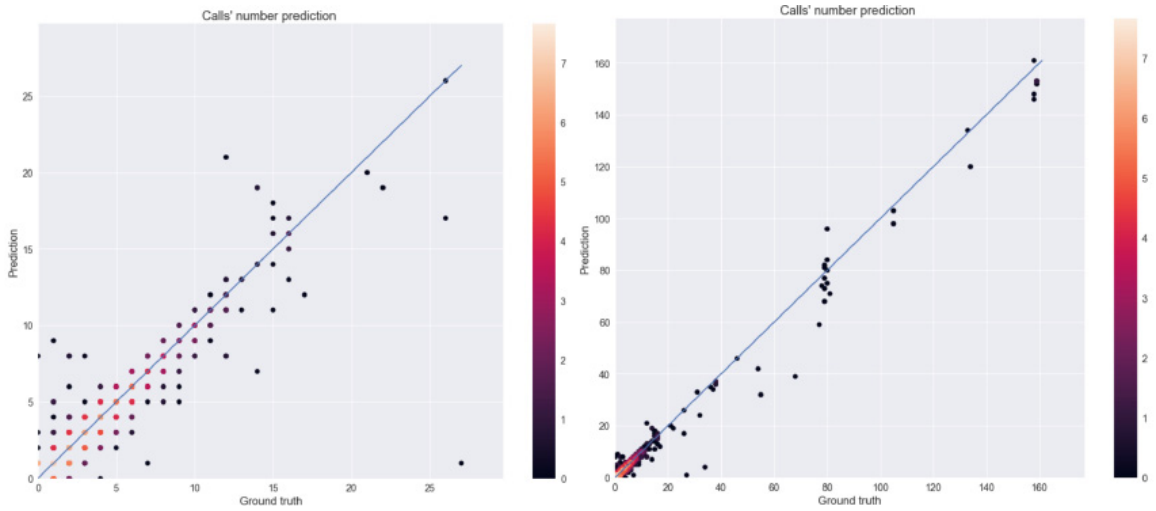

Figure 9: The results of the prediction of the activity of calls with bad termination codes in one of the clusters for a user. The x-axes shows the ground truth values and the y-axes shows the predicted values. The blue line shows the ideal case, when the predictions are exactly the same as the original values. The colors of the dots correspond to the number of the points with approximate corresponding values. The color bar is in logarithmic scale. That means that, for instance, the light orange dot says that there are ef points with the corresponding x- and y-axes values.

Because one goal is to avoid generating too many false positives, there is focus on what to do in case of high values of the anomaly score s_t. First, "suspiciously high" will be defined. Most algorithms for anomaly detection have the tendency to generate alerts if single data samples are above some internal thresholds used by the algorithms. This behavior is referred to as noise susceptible. In VCAD, it is desirable to avoid trigger alerts based on isolated divergences between the prediction and empirical observations, and instead only trigger if those divergences occur and persist for prolonged times. If successful,

a system has been designed which is noise-robust and hence ready for operational environments. To achieve the objective, the distribution of the anomaly scores is built to verify the likelihood that a given current state is indeed anomalous.

A fixed-length rolling time window W is considered which captures the last N values of the computed anomaly scores before the current time. The rolling normal distribution is modeled. The mean μ_t and the variance σ_t^2 of the distribution for each rolling window W are then computed, as shown in Equations 10 and 11:

$$\mu_t = \frac{\Sigma_{i=0}^{W-1} s_{t-i}}{W}, \qquad \text{Equation: 10}$$

$$\sigma_t^2 = \frac{\Sigma_{i=0}^{W-1} (s_{t-i} - \mu_t)^2}{W-1} \qquad \text{Equation: 11}$$

The corresponding Q-function is considered to compute the Gaussian tail probability. Q(x) for a variable x extracted from a normal distribution is the probability that the variable takes a value greater than x.

Let $\tilde{w}$ be a time window for short term moving average, $\tilde{w} << W$. The likelihood score at the moment t, L_t may be computed as:

$$L_t = 1 - Q\left(\frac{\tilde{\mu}_t}{\sigma_t} - \frac{\mu_t}{\sigma_t}\right), \qquad \text{Equation 12}$$

where the mean is:

$$\tilde{\mu}_t = \frac{\Sigma_{i=0}^{\tilde{W}-1} s_{t-i}}{\tilde{W}} \qquad \text{Equation 13}$$

Now, any data sample may be scored more reliably by thresholding the likelihood score. By defining the anomaly susceptibility as ε (it is recommended to set it to a very small number (i.e., ε ~ 0)), then a moment t is considered to be anomalous if Equation 14 holds true:

$$L_t > 1 - \tilde{\varepsilon}, \qquad \text{Equation 14}$$

In the case of "well-behaving" predictable scenarios, $L_t \sim s_t$ (the behaviors are similar). Also, a single spike in s_t may not lead to a spike in L_t and hence VCAD may not triggered on isolated divergences. Conversely, a series of consecutive divergences may,

which means an anomaly detection system which is robust and resilience to noise. VCAD may achieve an accurate model that is resilient to noise prediction. VCAD uses the models and techniques described herein to detect anomalies in the past and use this knowledge to predict future states and alert whether future anomalies have a highly likelihood to happen in the future.

The engineering features and models used by VCAD may be used to predict the likelihood that anomalies will happen at future times. It does so using independent models (models which consider each feature in isolation which is analyzed as a univariate time series) and a multivariate model (models which consider all available features together and is analyzed as a multivariate time series).

A special classifier may consume as an input the output scores of all models and produce as the output the single VCAD decision on whether a time moment is anomalous or normal. M models may run in parallel, each producing its anomaly score for every time moment. By assuming that the models are all independent from each other, the joint distribution may be computed for any time moment as:

$$P(s_t^0, \dots, s_t^{M-1}) = \prod_{i=0}^{M-1} Q\left(\frac{\tilde{\mu}_t^i - \mu_t^i}{\sigma_t^i}\right), \quad \text{Equation:15}$$

where the variables are computed by each model in isolation using Equations 10 and 11. The joint anomaly likelihood score can then be computed by generalizing Equation 12 as follows:

$$L_t = 1 - \prod_{i=0}^{M-1} Q\left(\frac{\tilde{\mu}_t^i - \mu_t^i}{\sigma_t^i}\right) \quad \text{Equation: 16}$$

Similar to operations performed for each model in isolation, the joint anomaly susceptibility score may be derived, and anomalies may be triggered by thresholding its value (generalization of Equation 14).

The User Interface component of the VCAD system is now described in greater detail. It includes reports which can be consumed by domain experts to (i) view temporal trends and statistics about voice behaviors, or (ii) drill down into abnormal events for further investigation and remediation, and (iii) annotate events (upon the completion of their investigation) and hence use the domain-expert provided feedback as a way for

VCAD to improve the accuracy and precision of its predictions over time (semi-supervised modality of operation of VCAD).

Figure 10 below illustrates a first example of reports used by domain experts to monitor the behavior of the voice service and to promptly be notified if an anomalous event has been detected by VCAD. The voice behavior is shown at the trunk/gateway level. Also shown is the resource utilization trending as well as the associated port utilization and call activities. This view may be easily customized (adding new viewpoints) and tailored to the needs of operational personnel and users. The observed activity is represented by the blue lines, the predicted/expected behavior by VCAD is represented by red lines. Any anomalous event (i.e., divergence between observed and predicted values) is shown as the corresponding anomaly.

Figure 10: VCAD User Interface - Trends. Number of calls from a user on a selected trunk. This view shows the behavior over a one-month time window. The observed values are reported in blue, the predicted value reported in red, and if a moment t is anomalous, it is highlighted with a circle (on the top) and associated anomaly identifier. An anomaly susceptibility is chosen relatively high for this case.

Figure 11 below illustrates a second example of reports used by domain experts to monitor the behavior of the voice service and to promptly be notified if an anomalous event has been detected by VCAD. As shown, the user may simply click on an anomalous event to drill down for further investigation. Domain experts may easily log their feedback about an anomalous event after their investigation. The outcome of their investigation is then absorbed by the VCAD backend system and leveraged to recalibrate their runtime models to further improve the predictions (e.g., reduction in the false positive rate).

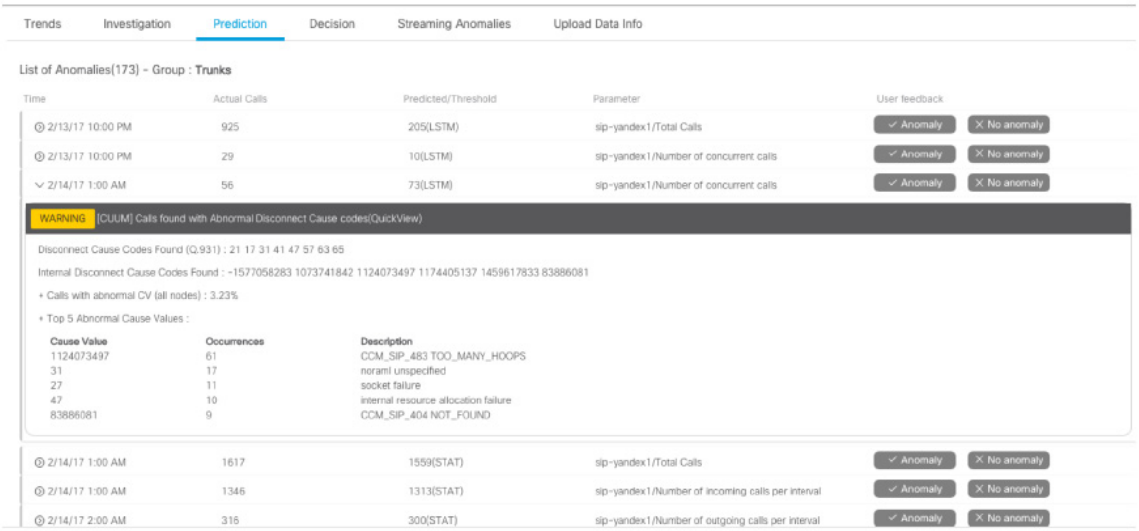

Figure 11: VCAD User Interface – Prediction and Annotation. Any anomalous event reported in the view Trends can be further investigated by domain expert using this VCAS UI view. When investigation is over, domain expert can simply label the suspicious event as normal or anomaly. This feedback will be leveraged by VCAD Model Generation components to recalibrate the models to improved future predictions the predicted value reported in red, and if a moment t is anomalous, it is highlighted with a circle (on the top) and associated anomaly identifier.

Before explaining how the feedback is actually used by the scoring models, the process of annotation and labeling of suspicious events is not mandatory for VCAD, meaning VCAD may operate in a pure unsupervised fashion. At the same time, VCAD's flexible design allows domain expert feedback to be taken into account during the process when possible to further improve its underlying predictive models and hence operate in a more rigorous semi-supervised mode.

For scoring the trained models, the test dataset may be obtained. In the supervised case, some number of the historic annotated data may be taken from the experts' feedback. In the unsupervised case, such labeled data may be generated. The diapasons where anomaly scores are very low (i.e., the feature values behave normally) may be identified. Some values a_t from these diapasons may be increased:

$$a'_t = a_t + b + r, \qquad \text{Equation: 17}$$

where a_t is the actual value for the time t, b is the threshold for the difference between the actual and predicted values, and r is some random value, $r \in (0, r')$, where r' is the maximum among residuals between actual a_t and predicted value f_t for all historical data T:

$$r' = \max_{\{t \in T\}}(|f_t - a_t|), \quad \text{Equation: 18}$$

Here, a'$_t$ becomes the fake generated anomalous point and it is labeled correspondingly.

At this point, labeled data has been obtained. F_1 measure to compute the score of a model:

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \quad \text{Equation: 19}$$

where precision and recall can be calculated as:

$$\text{precision} = \frac{TPR}{TPR + FPR},$$

$$\text{recall} = \frac{TPR}{TPR + FNR}, \quad \text{Equation: 20}$$

In Equations 19 and 20, True Positive Rate (TPR) is the number of correct positive results returned by a classifier, False Positive Rate (FPR) is the number of results mistakenly decided by a classifier as related to the positive class, and False Negative Rate (FNR) is the number of results mistakenly decided by a classifier as related to the negative class. The F_1 measure (Equation 19) captures the trade-off between precision and recall, with its best value being $F_1 = 1$ and worst value being $F_1 = 0$.

Figure 12 below illustrates an example where the anomaly scores are very low (i.e., the feature values behave normally).

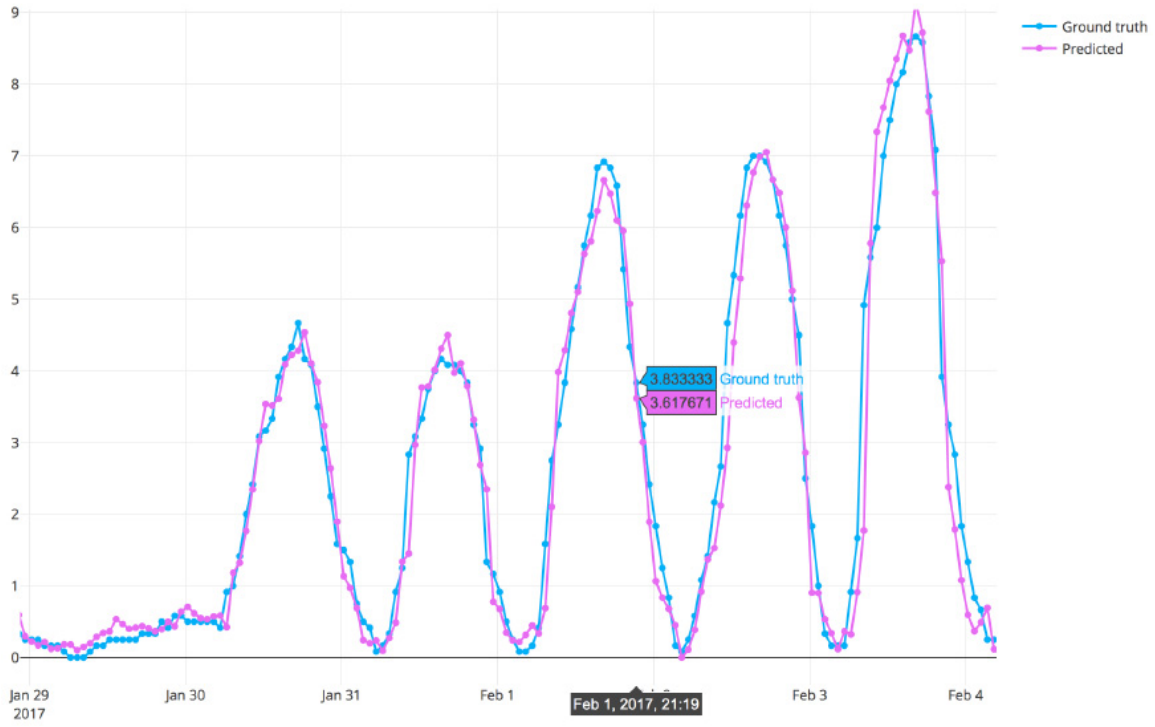

Figure 12: The number of calls integrated by five-minutes bins for a user for time diapason where the anomaly scores are very low – the feature values here are assumed to be normal (not anomalous).

Some illustrative examples of the VCAD system in action are provided using real data collected from real users. VCAD was applied on voice call log data (CDR/CMR records) collected from three large enterprises located around the world.

The first example involves user A. User A had two significant events, the outcome of which the VCAD system would have helped to mitigate. User A had encountered a fraud issue at a remote branch. Hackers had penetrated its telephony system undetected for nearly a month, and were making costly long-distance calls totaling over $800,000 USD. The pattern with such hackers is that they do not try to hide. Moreover, they steal as much as they can quickly because they know it will only be a matter of time before they are identified and shut down.

User A only became aware of the incident after they received a massive bill from their service provider. Had anomaly detection been in place at the customer site, it would have quickly identified and reported this anomaly to support personnel for further analysis and remediation.

When the VCAD system was applied, subject matter experts were able to identify the problem in an hour. Figure 13 below illustrates the report that the VCAD system generated for User A with details about the detected fraudulent activity.

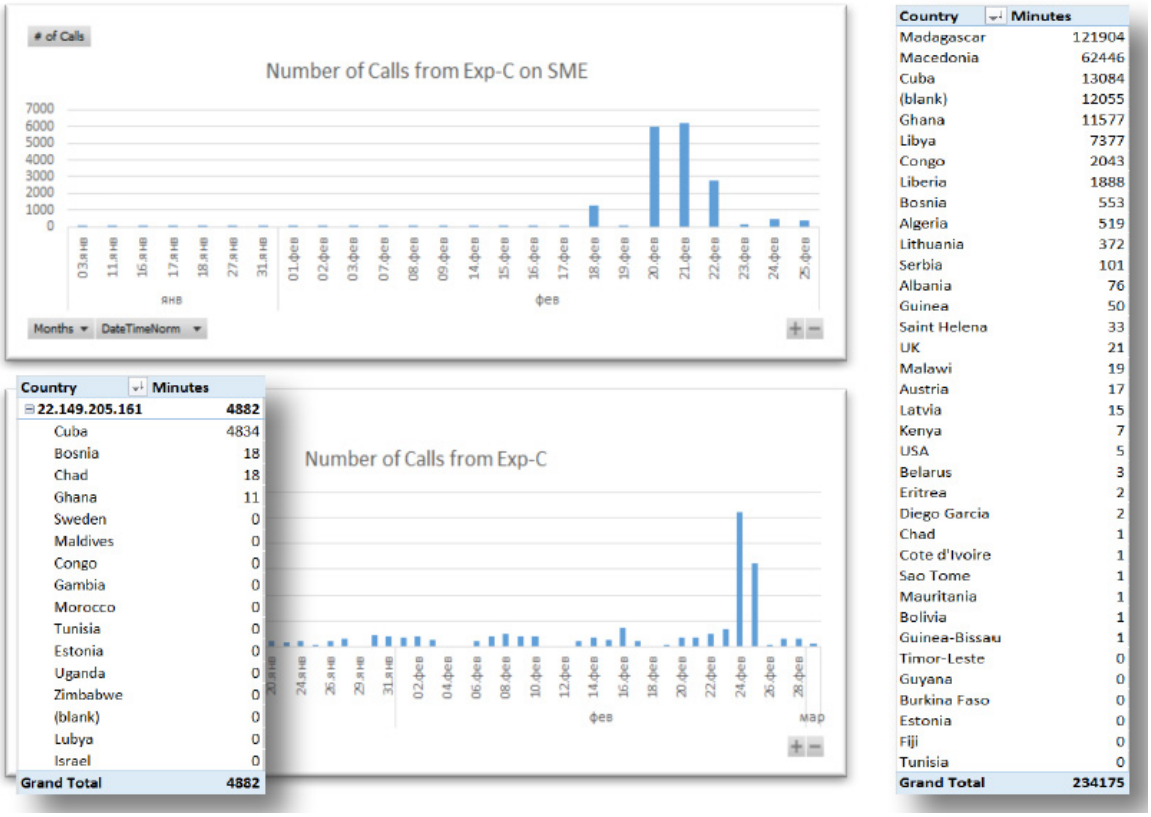

Figure 13: VCAD detected the excessive fraudulent activity User A experienced that went on for a month unnoticed. Reviewing the historical data, the VCAD system identified this inconsistent call activity within an hour highlighting also the countries where the calls were made to and from and the associated volumes.

Another event that User A experienced was in relation to their call service. User A was intermittently experiencing random poor voice call quality, resulting in call termination, but was unable to identify any consistency with call drops by location or by types of calls terminated. By analyzing historical data, the VCAD system revealed that the actual issue was with the end user devices. Defects in the phone's firmware specific to several models were causing the voice call quality issue. This again had been impacting User A's customer experience.

A second example involves User B. User B began experiencing call issues manifesting by random call drops with no specific visible pattern. It took this customer almost an entire day to identify and remediate the issue. The problem was traced back to network conditions that were causing media resource starvation/ failure. As illustrated in

Figure 14 below, the VCAD system processed User B's historical data and predicted the problem six hours before it happened (i.e., the yellow alerts). Then it started generating alerts of higher severity a few minutes before the problem started (i.e., the red lines). Without a system like VCAD, these kinds of issues can go on indefinitely, negatively impacting call quality and customer experience.

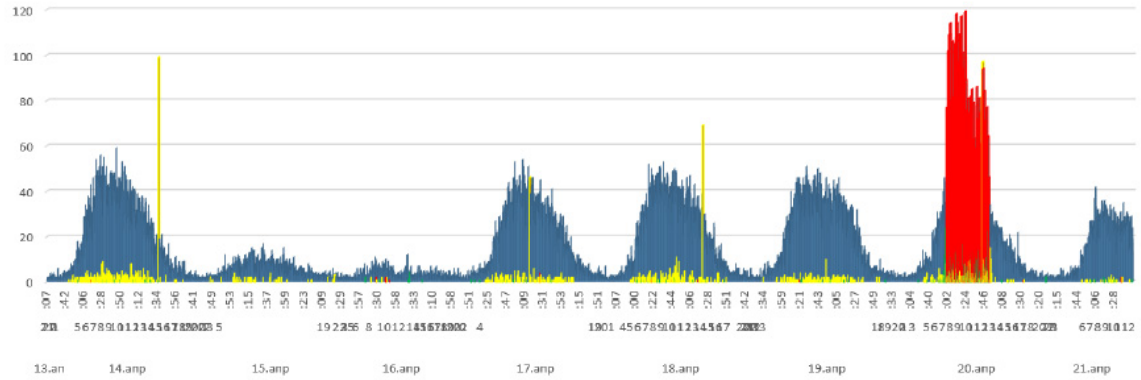

Figure 14: VCAD detected both the preliminary symptoms of the random call drops experienced by User B six days in advance (yellow lines) and detected high severity anomalies just a few minutes before the actual drops start happening (red lines).

The third example is with User C. At one point, User C's call center went down for a full ten minutes. The only way it was able to restore its service was to completely reboot its system. As illustrated in Figure 15 below, the benefits of the VCAD system, which enables data to be analyzed by multiple models simultaneously, would have proven invaluable in this case.

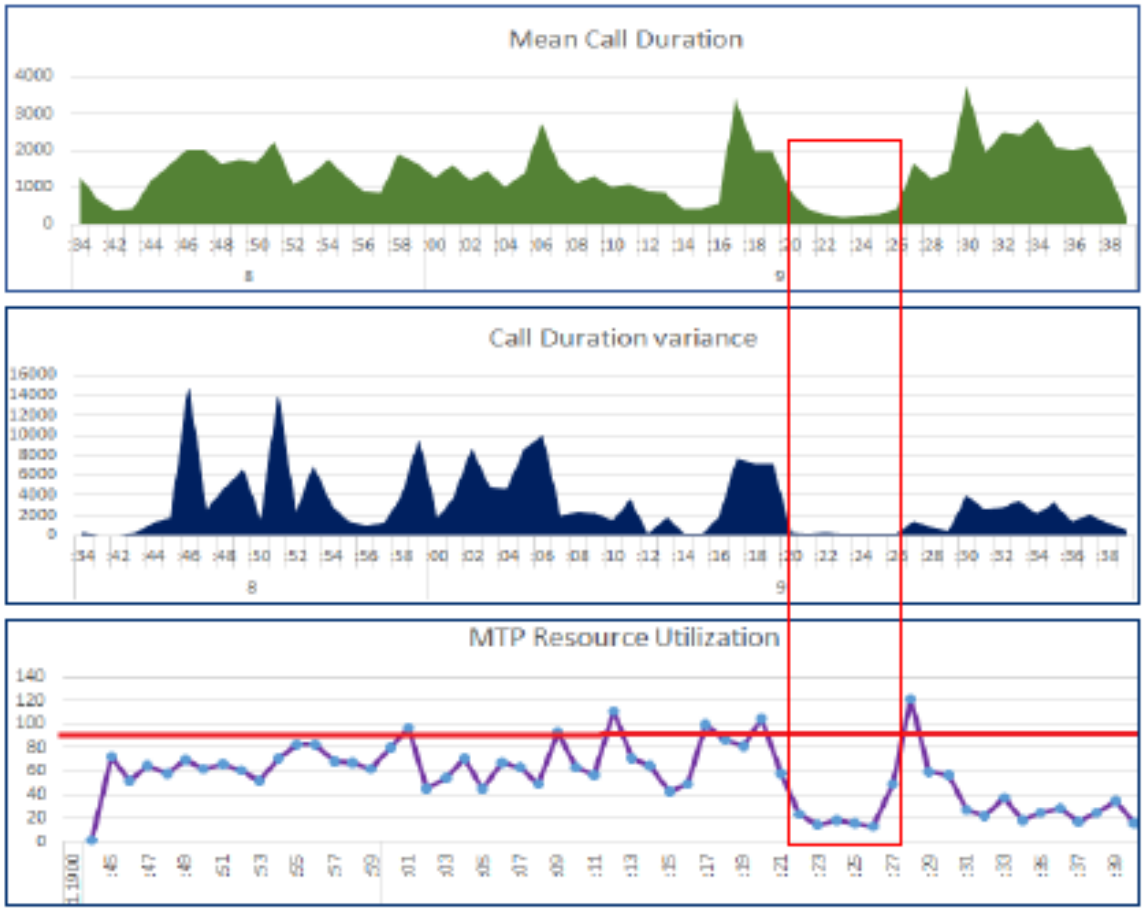

Figure 15: VCAD is able to analyze multiple layers of data and able to identify several problems. In the bottom graph, the horizontal red line indicates the defined demand on the system; the peaks above it demonstrate that it was more than resources available to meet that demand which explained the ten minutes of down-time that followed.

Further analysis of historical data determined the cause for the call center going down was due to several factors such as Interactive Voice Response (IVR) ports starvation and IVR script problems. Another issue identified was that they were exceeding their resource limits. The resources required to maintain service was simply were not there. This is an issue that could only be detected by analyzing data from multiple data sources.

Algorithms are provided that can detect and recognize patterns, and subsequent anomalies in those patterns. Since these anomaly detection models are applied to different elements of datasets, algorithms are able view the system state from many different angles, enabling the model to detect problems that may be occurring as the result of several factors. By detecting inconsistencies in patterns, these models may detect and alert parties of an unusual system behavior that would otherwise impact, and possibly disrupt, service. In some cases, anomaly detection may predict and prevent issues as well. These models

analyze data and provide contextual information about the problem type, timeframe, problem scope and even device, so engineers are able to rapidly pinpoint costly service issues for customers and provide fast resolution.

VCAD is an orchestration of advanced machine learning pipeline using LSTM and statistical modelling. Data may be collected real-time every five or ten minutes (although depending on the particular user this time can be adjusted) and predict six values for the next twelve points (sixty or one-hundred and twenty minutes, although this time can be adjusted as well) with 90% accuracy. Then users may be alerted if there is any anomaly in the actual value compared to the predicted values. The VCAD methodology may be applicable to call data from any call management systems which have CDR and CMR data. This methodology improves model accuracy by learning weights from real-time CDR/CMR data and dynamically adjusting the initial Neural Network (NN) model.

Figure 16 below illustrates an example overview diagram.

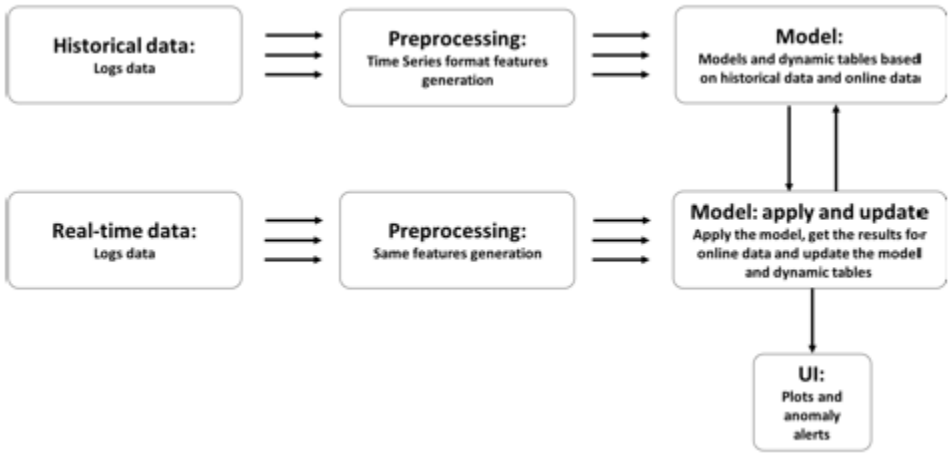

Figure 16

VCAD involves a method of learning the weights for later values rather than using fixed coefficients. The vector size of the weights may be chosen based on how many historical points to be considered. The fixed "intuitive" constant weights perform poorly for this data. These weights are provided as attention coefficients learned from CDR/CMR data by minimizing the loss function.

The anomalies are not triggered based only on the "standard" values. Meanwhile, the common method for detecting anomalies is taking moving averages/rolling means (they are different from the "standard" values) and some deviation value from it. "Standard"

values are used for understanding the range where the original value is for the time stamp. The goal is to predict whether it is anomalous, and then use predicted value and the difference between actual and possible original values for the time stamp in the future to explore with further statistical techniques whether the anomaly is coming.

Using only moving averages may enable detecting spatial anomalies (when the value is outside the typical range) but not temporal anomalies (when the value is not outside the typical range but the sequence in which it occurs is unusual). Since the VCAD algorithm combines statistical and deep learning algorithms, it can detect both.

VCAD is an automated and unsupervised algorithm which ingests, cleanses and profiles voice call logs in real-time. It continuously learns the behavior of a voice service and accurately tracks its evolution. It then uses historical data and advanced machine learning algorithms to identify early symptoms leading to future problems. The model is universal to process data in real-time, and learn while simultaneously making predictions. Though the methodology has been demonstrated by taking a few use cases in voice call log streaming data, it can be used for solving many anomaly detection problems for other network or system log data.

In summary, VCAD is described herein to detect inconsistencies in patterns. VCAD is an anomaly detection system which is based on a LSTM algorithm and statistical methods. By detecting inconsistencies in patterns, the models described herein may detect and alert user of unusual voice service behavior that if not properly corrected can degrade, and possibly disrupt, the voice service. The statistical and machine learning methods used by VCAD are generic and may be used for solving other time-series problems when using other type of logs such as call logs, game logs, application usage logs, etc. The VCAD proactive, predictive capabilities allow customers to either eliminate the issue altogether, or turn costly, unplanned outages into controlled maintenance windows.