

# Different every time: A framework to model real-time instant message conversations

Jonathan Dunne  
Hamilton Institute  
Maynooth University  
Email: jonathan.dunne.2015@mumail.ie

David Malone  
Hamilton Institute  
Maynooth University  
Email: david.malone@nuim.ie

**Abstract**—As startups and micro teams adopt real-time collaborative instant messaging solutions, a wealth of data is generated from day to day usage. Making sense of this data can be a challenge to teams, given the lack of inbuilt analytical tooling. In this study we model the distributions of duration, inter-arrival time, word count and user count of real-time electronic chat conversations in a framework, where these distributions can be used as an analogue to service time estimation of problem determination. Using both an enterprise and open-source datasets, we answer the question of what distribution family and fitting techniques can be used to adequately model real-time chat conversations. Our framework can help startups and micro teams alike to effectively model their real-time chat conversations to allow high value decisions to be made based on their collaboration outputs.

## I. INTRODUCTION

Real-time collaboration solutions are being marketed as a way for teams regardless of size of to increase their productivity [1] [2] [3]. One of the benefits of using such software is that conversations are segmented into either spaces, channels, or chat rooms which facilitate discussion in a linear fashion. As all conversations are recorded, rolling back to prior conversations can be done with the spin of a mouse wheel or the swipe of a screen. High end collaboration suite also include an additional set of features such as a file repository, knowledge management software and ability to screen share. A number of feature rich solutions include: Watson Workspace [4], Slack [5], Microsoft Teams [6] and Azendoo [7], to name but a few.

One of the key selling points of real-time collaboration suites is the idea that real-communication reduces the need for email communication [8], thus solving the problem of ‘email paralysis’ [9] which is the effect of having such a large volume of email an individual is unable to communicate due to the sheer volume of messages. However both micro-teams and startups face new challenges with the adoption of real-time collaboration software. As usage increases over time, so does the volume of data. Furthermore as current offerings offer little in the way of in-built analytical solutions, making sense of the growing volumes of collaboration data is key.

While micro-teams and startups have a number of key usecases, a growing trend is for development teams and DevOps alike to use real-time collaboration software to facilitate their ability to debug problems, otherwise known as problem determination [10] [11]. The time to debug and fix a problem is typically defined as the service time and the time between

successive problems is known as the inter-arrival time [12]. By a logical extension we can see that the duration of a group chat conversation and the time between the start of such conversations could be referred as an analogue to service time and inter-arrival time duration respectively. Modelling such data may give us greater insight in a teams ability to solve problems.

In this paper we propose a framework that both startups and micro teams can use to effectively model their group chat instant messaging conversations using a number of available techniques. The core idea of this framework is for small teams to use the output of modelled conversations to gain insight into the expected time of a group chat and once a conversation has completed when the mean time until the next conversation begins. For startups and micro teams with a limited team size, understanding the duration of a group chat conversation can aid problem resolution outcomes.

This study contains research conducted on two large real-time chat discourse datasets. Our first dataset is an enterprise dataset from a real-time collaboration application, our second dataset is an open source data set from an internet chat relay (IRC) channel. Through study of this data we investigate what techniques can be employed to effectively model the distributions of chat duration, interval, inter-arrival time, the number of words per chat conversation and the number of users per single chat conversation. Using the results of this study for our framework, a modelling suite can be developed to provide teams with a greater level of introspection of their chat data.

The rest of the paper is structured in five sections: Section II provides a description of background and related works. Section III describes the both datasets as well as our method and approach. Section IV provides analysis of our experiments. It is followed by section V that explains our results. Finally, the conclusion and future work are described in section VI.

## II. BACKGROUND AND RELATED RESEARCH

### A. Distribution Fitting

Probability distribution fitting is the fitting of a known probability distribution to a data set regarding the repeated measurement of a variable phenomenon. The type of fitted distribution can vary depending on the under lying data set. The main purpose of distribution fitting is to predict the probability or to forecast the frequency of occurrence of the magnitude of the phenomenon in a certain interval.

There are two main fitting techniques used. The first method is called the method of moments. This method uses expected values of a random variable (a moment) from a population. A sample is then taken from the population and subsequent moment is estimated. The sample moments are used to make estimates about an unknown population. This idea was first proposed by Karl Pearson in 1894 [13].

The second method is called Maximum likelihood estimation (MLE). MLE is a method to estimate the parameter values of a model by determining parameter values that maximise the likelihood. This method was first proposed by Ronald Fisher in the 1920's [14], with a subsequent formal proof by Samuel Wilks in 1938 [15].

### *B. Goodness of Fit Testing*

If a suitable probability distribution can be found to fit a data set, of interest is how well the distribution fit that data. A number of methods have been developed to assess the goodness of fit of a distribution to a data set. We shall discuss three of the main tests briefly.

The Cramér-von Mises criterion [16] [17] is a non-parametric test which examines the goodness of fit of a cumulative distribution function (CDF) compared to that of an empirical density function (EMF). Using a significance test we can test a hypothesis of whether a data set is drawn from a given probability distribution

The Kolmogorov-Smirnov [18] test quantifies a distance between the an EMF of the sample and the CDF of the reference distribution, or between the EMF of two samples. The idea being that the closer the distance between the two, the better the fit.

The Anderson-Darling [19] [20] test is a statistical test of whether a given sample of data is drawn from a given probability distribution. This test is a modification of the Kolmogorov-Smirnov test as it gives more weight to the tails of data.

### *C. Heavy Tailed Estimation*

In probability theory, heavy-tailed distributions are distributions whose tails are not exponentially bounded. In fact these distributions have much heavier tails, for example a Pareto or Generalised extreme value distribution. For such distributions a tail index, which is essentially the shape parameter of a distribution can be used to make inferences about the underlying data.

Hill [21] proposes one of the first methods to infer tail behaviour of a distribution function. This work is valuable in that no prior assumption of the type of distribution is required prior to inference. His tail estimation technique is one of the standard methods for measuring the index of a heavy-tailed distribution.

Pickands [22] provides a method to make inferences about the tail of a probability distribution function. This method can be applied to all continuous distribution functions. Pickands method is an alternative method to calculate the index of a heavy-tailed distribution.

Nair et al. [23] discuss the idea that heavy-tailed data and their corresponding distributions are a more common occurrence. They also discuss various techniques to model distributions from heavy-tailed datasets.

### *D. Hurdle Distribution*

Hurdle distributions are a class of distributions for count data that can help manage data sets with a large number of zeros or a count dataset which exhibits either over-dispersion or under-dispersion. Mullahy [24] proposes the idea of a hurdle model which provides a more natural means to model over or under-dispersed count data.

### *E. Kernel Density Estimation*

For datasets which do not fit a known distribution family, a non-parametric approach can be taken. One such approach is Kernel Density Estimation (KDE). In KDE, a range of kernel (weighting) functions are applied to a dataset plotted as a histogram. The kernel functions are divided into various widths (bandwidth). The goal is to choose the most appropriate kernel bandwidth and function shape that best fits the histogram. Both Rosenblatt [25] and Parzen [26] are credited with creating KDE in its current form.

A number of significant contributions in this area have also been made in the field of KDE. These are discussed briefly.

Kernel performance is measured by either the mean integrated squared error (MISE) or the asymptotic mean integrated squared error (AMISE). Epanechnikov [27], proposed a parabolic shaped kernel that minimises AMISE and is therefore optimal. Kernel efficiency is now measured in comparison to the Epanechnikov kernel.

Silverman [28] proposes an improved method for bandwidth selection. In his study, if a Gaussian basis function is used to approximate univariate data, and if the underlying density is Gaussian, the optimal choice for the bandwidth parameter is the standard deviation of the samples. This method is known as Silverman's rule of thumb or the Gaussian approximation.

Sheather and Jones [29] provided an improved method for data-based selection of the bandwidth in KDE. Their paper included a new bias term in their bandwidth estimate, that provides good performance for a broad set of cases.

### *F. Other related studies*

Dewes et al. [30] conducted a study to better understand network traffic dynamics by examining Internet chat systems. While their main research output was to demonstrate how to separate chat traffic from other Internet traffic, the authors conducted analysis of the inter-arrival times of chat messages. The authors hypothesis was as follows: Are the inter-arrival times of chat messages consistent with an exponential distribution? The hypothesis was rejected due to lack of evidence, however they found the inter-arrival times were more consistent with a heavy-tailed distribution.

Lukasik et al. [31] modelled time series data of tweets to understand if a reliable prediction model could be derived to predict future tweets. Their research found that by employing

a log-Gaussian Cox process a higher degree of predictive precision could be achieved. The authors also found that mining text from tweet messages can improve inter-arrival time prediction.

Vande Kerckhove et al. [32] provided research into the field of inter-arrival times of electronic communication. The authors investigated the level of inter-event dependence between postings and whether a Markovian process would be suitable to model the memory effect observed in inter-arrival online activities. For their study the authors social media data from Twitter and Reddit. Their research concluded that by allowing dependence between message wait times allows for more precise modelling than by fitting against a power-law distribution alone.

Markovitch and Krieger [33] compared the nonparametric estimation of the probability density function of long-tailed distributions from internet based traffic against existing parametric methods. The authors found that a neither a Pareto nor an exponential model was a suitable fit to their underlying data. Additionally by using both a Parzen–Rosenblatt kernel and a histogram of variable width (a polygram) a more suitable fit was achieved.

Maieroda and Markovitch [34] discuss the nonparametric estimation of a heavy-tailed probability density function by a variable bandwidth kernel estimator. The authors discuss two approaches: A preliminary transformation to provide an information estimation of tail density and a the discrepancy method based on the Kolmogorov–Smirnov statistic to evaluate the bandwidth of the kernel estimator. The authors use internet based traffic to validate their models.

Wang [35] presents a how-to article on visualising the inter-arrival times of tweets. Using the R programming language the author describes the process to collect, visualise and determining if the inter-arrival times can be modelled by a Poisson process.

### III. DATA SET

Inter-arrival time modelling of social and collaboration message data has been shown to provide a useful way to make inferences about the underlying structure of message data. By combining prior research with our proposed framework can allow startups and micro-teams to infer the expected duration of chat conversations. This output can effective analogue to service time determination.

The study presented in this paper examines approximately 540 real-time chat conversations from two datasets. A number of points related to each dataset are summarised in Table I.

The first dataset analysed was the open source Ubuntu dev IRC channel. [38] For our study we reviewed approximately 4200 messages. For each message we reviewed whether a subsequent message was part of an existing conversation or part of a prior or subsequent conversation. For each unique conversation identified we assigned a numeric topic ID. As part of the review phase we annotated 231 unique conversations. The total time period analysed was approximately 86 hours.

The second dataset analysed was from enterprise instant message chat system which discussed cloud infrastructure

problems. For our study we reviewed approximately 3200 messages. For each message we reviewed whether a subsequent message was part of an existing conversation or part of a prior or subsequent conversation. For each unique conversation identified we assigned a numeric topic ID. As part of the review phase we annotated 312 unique conversations. The total time period analysed was approximately 4820 hours.

In an ideal world a chat conversation will start, progress then reach a logical conclusion, however on occasion an unrelated message will be injected into an existing chat conversation. We found a number of heterogeneous chat messages which appeared mid way through a homogeneous chat conversation. We enumerated these ‘entangled chat conversations’ [39] in total 57 of the chat conversations from the Ubuntu IRC dataset and 27 conversations from the enterprise dataset were found to be entangled. It should be noted that chat disentanglement is beyond the scope of our study and will not be discussed further.

This study aims to answer the following questions. First, can the duration of our annotated chat conversations be modelled by a parametric method? If not can a non-parametric method be used? Second, can the durations between annotated chat conversations be modelled by a parametric method? If not can a non-parametric method be used? Third, what is the most appropriate method to model the inter-arrival times of chat conversations? Fourth, what modelling techniques can be used to model the number of words and lines of text in a chat conversation? Fifth, to model the number of users present in a chat conversation, is a Poisson model appropriate?

#### A. Conversation duration modelling

Conversation duration is defined as the timestamp of the last message in a conversation subtracted by the timestamp of the first message in a conversation. A number of conversations were recorded as being zero minutes in length, this is due to a number of short (five messages or less) conversations completing in less than one minute.

Measuring the conversation duration is useful exercise in that given many teams use real-time chat collaboration software to discuss and debug problems, we can use the conversation as an analogue to measure service times.

Probability distributions are used in many fields to understand how likely it is for an event to happen. In the case of chat conversation duration times, our starting point is to conduct a parametric test to determine if a known distribution can be fitted to our data set. The benefit of attempting to fit a known distribution is that, if such a fit can be found, we can gain access to the mathematical properties of such a distribution (i.e. mean, variance, probability density function, cumulative density function etc.).

When parametric methods fail to yield a useful result, additional methods are employed (i.e. Distribution body and tail modelling, Hurdle methods and non-parametric methods such as KDE)

For distribution fitting, we used the R package `fitdistrplus` [40] to fit various distributions to our dataset. To validate the efficacy of each distribution, the authors used the R package

TABLE I. SUMMARY OF DATASET METRICS AND FACTORS

Dataset	Total No. Messages	Duration (dd:hh:mm:ss)	Conversations Annotated	Mean No. Messages per Hour	No. Entangled Conversations	Entangled Conversation Ratio
Ubuntudev-IRC	4223	3 days, 14 hours, 15 mins 0 secs	231	49.10	57	0.25
Enterprise Instant Message chat system	3261	200 days, 21 hours, 38 mins and 53 secs	312	0.68	27	0.09

ADGofTest [41], which uses the Anderson-Darling goodness-of-fit test, to determine if the observed data follows a specific distribution [20].

### B. Conversation delta time modelling

Conversation delta time is defined as the time duration between chat conversations. For example the timestamp of the starting message in a second conversation is subtracted by the timestamp of an ending messaging in a first conversation. It should be noted in the case of an entangled conversation, a conversation delta is recorded by a negative time value. While this may seem counterintuitive, our reasoning is that while we had a mechanism to record the number of entangled conversations we also needed to measure the level of entanglement in terms of time. By using negative time we can in effect determine at the glance which conversations are entangled.

Measuring and modelling conversation delta times can surface the waiting time between prior and future discussions. These results can help answer questions around the expected time between conversations.

Due to the complex nature of the underlying data set (i.e. a mixture of logical conversation durations (positive durations) and entangled (negative durations)) Two approaches were considered. The first was to split the dataset into two smaller subsets, one subset contained the positive durations and second subset contained the negative durations. For distribution fitting we used the absolute values for the entangled conversation subset. The second approach was to conduct KDE modelling on the entire conversation duration dataset.

The approach to distribution fitting and validation is the same as described in the previous subsection. Our approach to conducting a non-parametric test (KDE) to model the entire delta duration (both logical and entangled) was conducted using the R package Density [42].

### C. Conversation inter-arrival time modelling

Conversation inter-arrival time is defined as the time duration between the start of a first chat conversation and the start of a second chat conversation. In other words, the inter-arrival time is essentially the product of conversation duration plus conversation delta time.

Measuring and modelling the inter-arrival time conversation is beneficial. The inter-arrival time is an important component, when combined with conversation duration modelling the result can be used for to predict conversation busy and free times as part of a wider queue framework.

The approach to distribution fitting and validation is the same as described in the previous subsections.

### D. Conversation message & word modelling

A key component of any group chat conversation are the number messages that are required to complete a conversation and the amount of words used. Performing analysis on both variables can initially tell us if a distribution can be fitted to the underlying data. If a suitable distribution can be found, this result can help answer questions such as the expected number of lines and words in a chat conversation.

Thereafter additional inference can be conducted such as topic and keyword analysis. However both topic and keyword analysis is beyond the scope of this current work and will be discussed in reference to future work in the conclusion.

Our approach to distribution fitting and validation is the same as described in the previous subsections.

### E. Conversation user count modelling

Conversation user count is defined as the number of unique users that contribute at least one message to a group chat conversation.

Like previous sections if a suitable distribution can be found to fit user count data, this result can assist teams in determining the expected number of participants per chat conversation or the proportion of conversations that contain n number of users.

As we are dealing with count data our initial approach will to determine if a Poisson distribution is a suitable fit to our user count data. If there is sufficient evidence to suggest a lack of fit, a test for over dispersion and under dispersion will be conducted. If there is evidence to suggest some level of dispersion within our data, we shall employ a method of hurdle modelling. This model will then be tested for goodness of fit.

To validate the goodness-of-fit of a Poisson distribution, the authors used the R package vcd [43], which uses the Chi-Squared goodness-of-fit test, to determine the level of dispersion in our count data we used the R package AER [44].

### F. Limitations of dataset

The dataset has a number of practical limitations, which are now discussed. The process of aggregating chat messages into a cohesive conversation is a subjective one. While every effort was made on the part of the authors to align messages to a thread we accept that the process is subjective. Additionally the post times for the Ubuntu chat were measured in hours and minutes only. As a result conversation duration, delta and inter-arrival times were recorded to the nearest minute, whereas for the enterprise data set, these times were recorded to the nearest second.

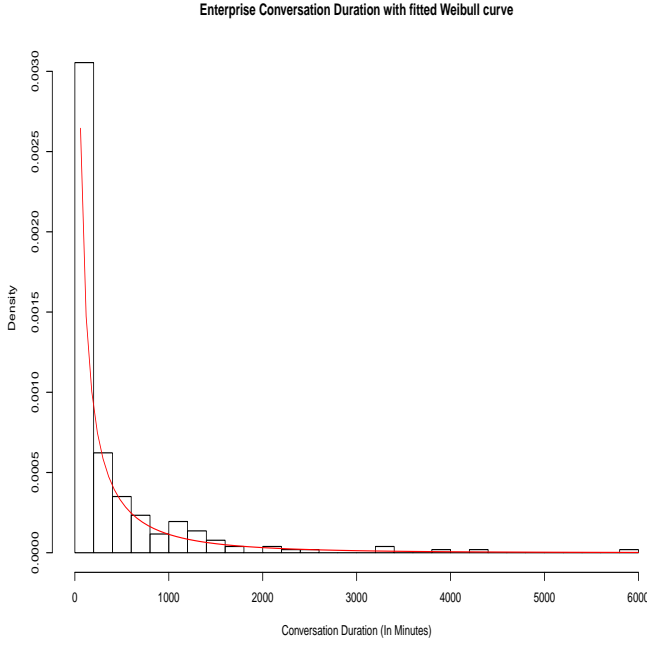


Fig. 1. Enterprise conversation duration with fitted Weibull curve

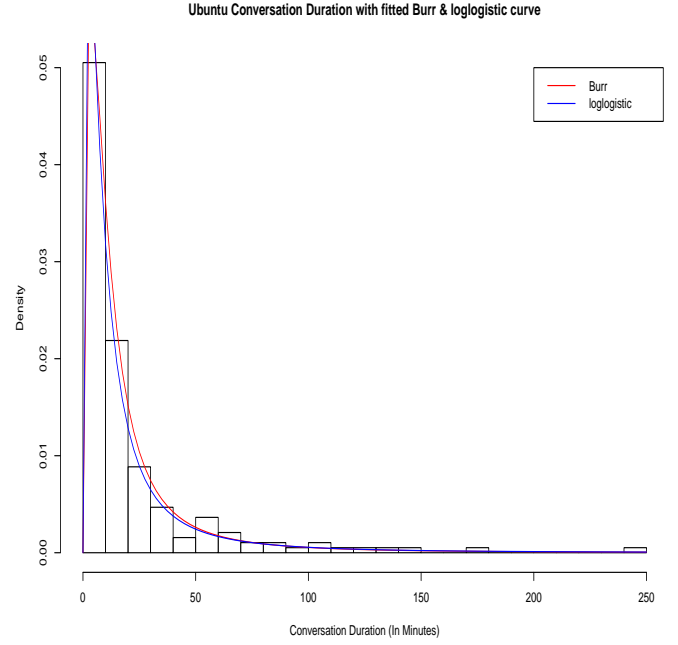


Fig. 2. Ubuntu conversation duration with fitted Burr loglogistic curve

The chat conversations that form part of this study are from a) a Ubuntu IRC developer channel and b) from an enterprise chat messaging system that discussed cloud infrastructure problems. As a result our analysis may not be relevant outside of these fields.

#### IV. RESULTS

We now explore the results of our analysis. A Table II contains a summary of the results for easy reference.

##### A. Conversation duration modelling

Fig. 1 shows a probability density function histogram for the enterprise dataset. A total of 55 conversations were found to be of 0 minutes in length (i.e. conversations that were started and completed in less than 1 minute). These values were removed from the dataset and a Weibull distribution was found to be best fit for the remaining 257 samples. An Anderson–Darling test statistic and p-value were computed as 1.1 and 0.31 respectively. The p-value is above the 0.05 significance interval.

Fig. 2 shows a probability density function histogram for the ubuntu dataset. For this dataset 39 conversations were found to be of 0 minutes in length. Once again these values were removed from the dataset. Both a Burr log-logistic distribution were found to be the best fit for the remaining 192 samples. An Anderson–Darling test statistic and p-value were computed for both distributions as 1.3 and 0.61 respectively. The p-value is above the 0.05 significance interval.

##### B. Conversation delta time modelling

The conversation delta time modelling results are split into two parts. The first is a parametric approach using MLE. In

this approach the conversations were divided into two subsets: logical conversations (i.e. time duration between the end of an  $n$ th and the start of an  $n+1$  conversation, which is positive), and entangled conversation (i.e. time between the end of an  $n$ th and the start of an  $n+1$  conversation, which is negative). The second approach is a non-parametric approach using KDE.

Fig. 3 and Fig. 4 show probability density function histograms of both the entangled and logical conversation delta times for the enterprise dataset. A Weibull distribution was found to be the best fit for both sub datasets. An Anderson–Darling test statistic and p-value was computed for both distributions as 0.49 & 0.76 (logical dataset) and 0.3 & 0.94 (entangled dataset). In both cases the p-value was found to be above the 0.05 significance interval.

Fig. 5 shows the output of a histogram of the combined entangled and logical conversation delta times for the enterprise data set. The Sheather–Jones direct plugin bandwidth selector combined with a uniform (rectangular) shaped kernel was found to be the optimal fit. The bandwidth was computed as 56.73.

Fig. 6 and Fig. 7 show probability density function histograms of both the entangled and logical conversation delta times for the Ubuntu dataset. A small transformation (1 minute) was added to the logical delta subset. A loglogistic distribution was found to be the best fit for both subsets. An Anderson–Darling test statistic and p-value were computed for both distributions as 2.46 & 0.052 (logical dataset) and 0.60 & 0.64 (entangled dataset). In both cases the p-value was found to be above the 0.05 significance interval. For the logical dataset we quote the p-value in this case to three decimal places to illustrate the p-value was calculated as above 0.05.

Fig. 8 shows the output of a histogram of the combined

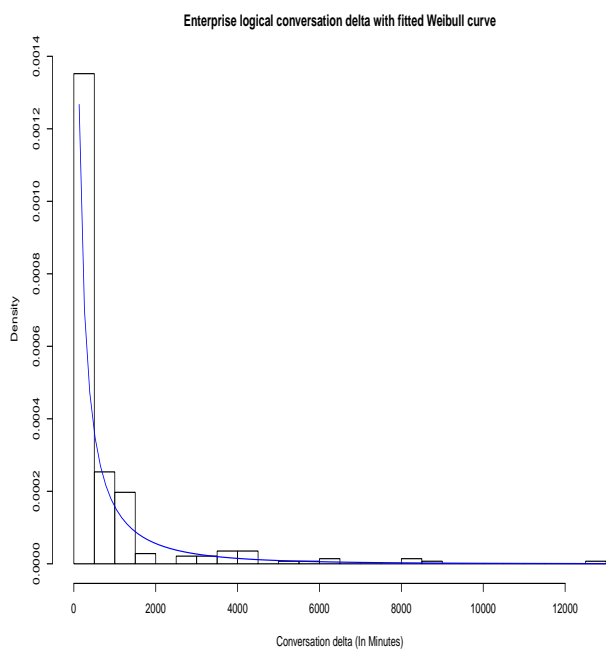


Fig. 3. Enterprise logical conversation delta, with fitted Weibull curve

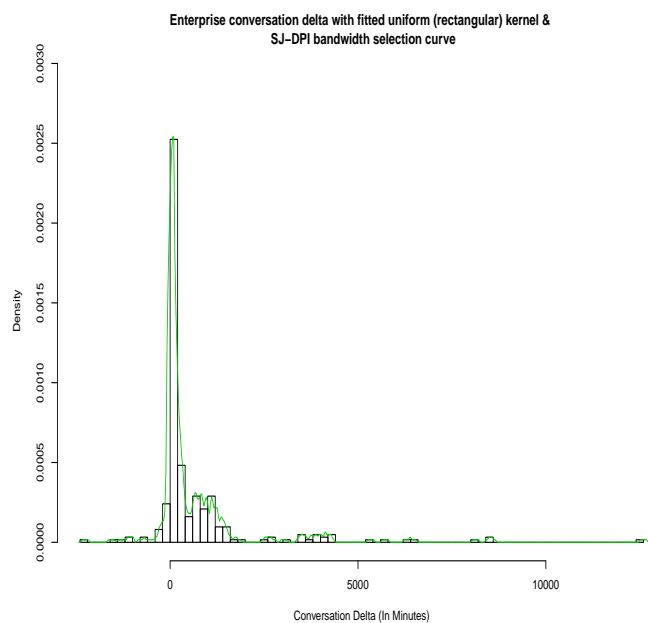


Fig. 5. Enterprise conversation delta with fitted uniform (rectangular) kernel SJ-DPI bandwidth selection curve

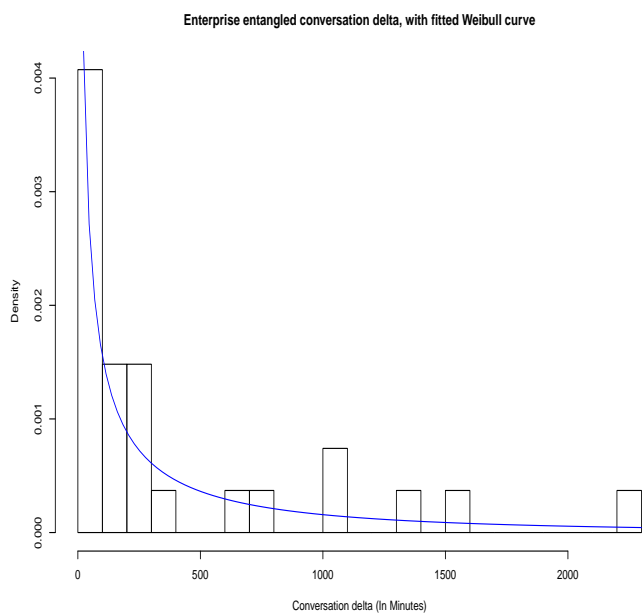


Fig. 4. Enterprise entangled conversation delta with fitted Weibull curve

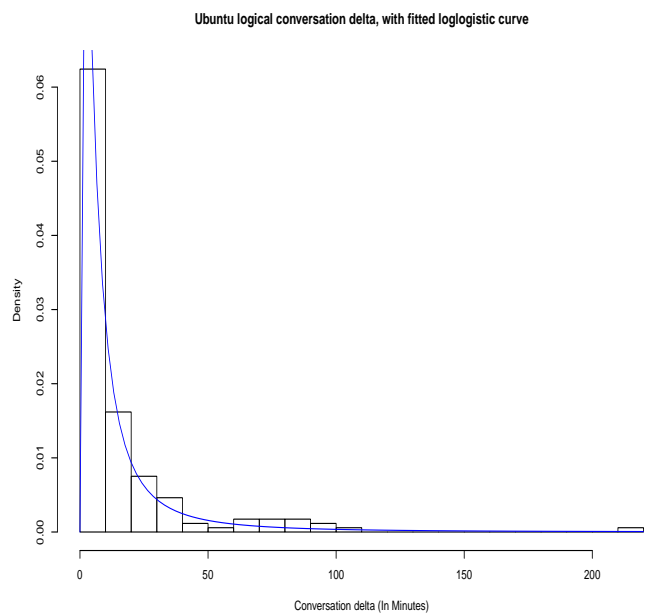


Fig. 6. Enterprise logical conversation delta, with fitted Weibull curve

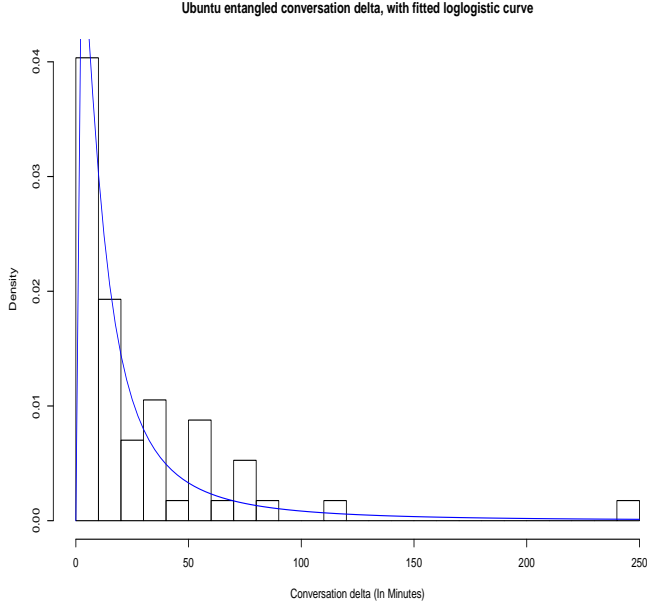


Fig. 7. Enterprise entangled conversation delta with fitted Weibull curve

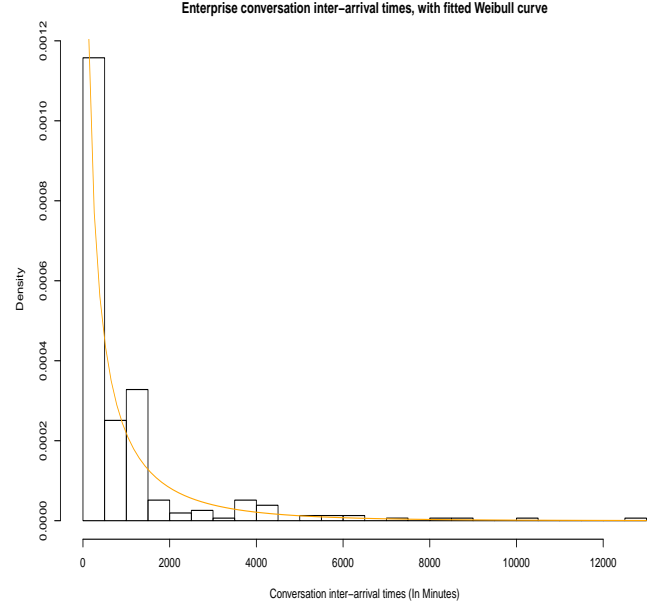


Fig. 9. Enterprise conversation inter-arrival times with fitted Weibull curve

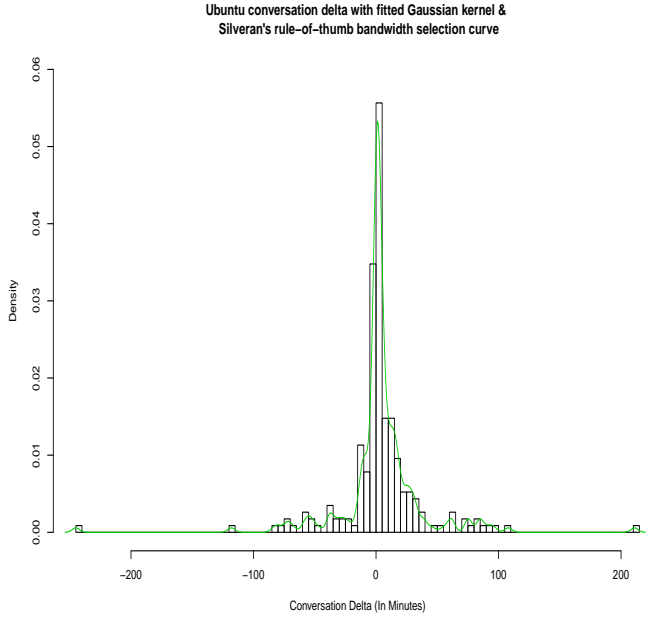


Fig. 8. Enterprise conversation delta with fitted uniform (rectangular) kernel SJ-DPI bandwidth selection curve

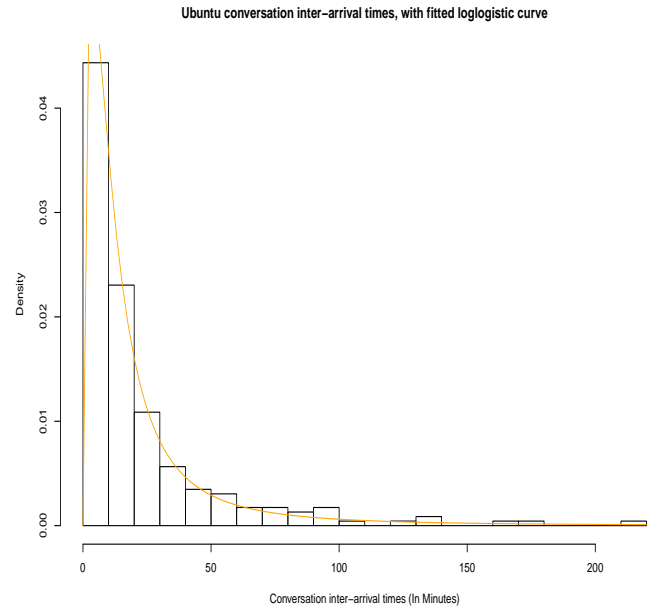


Fig. 10. Ubuntu conversation inter-arrival times with fitted loglogistic curve

entangled and logical conversation delta times for the Ubuntu data set. Silverman's rule-of-thumb bandwidth selector combined with a Gaussian shaped kernel was found to be the best fit. The bandwidth was computed as 2.94.

### C. Conversation inter-arrival time modelling

Fig. 9 shows a probability density function histogram for the enterprise dataset. A Weibull distribution was found to be best fit. An Anderson–Darling test statistic and p-value were

computed as 0.78 & 0.5 respectively. The p-value is above the 0.05 significance interval.

Fig. 10 illustrates a probability density function histogram for the enterprise dataset. A small constant (1 minute) was applied to each value in the dataset. A loglogistic distribution was found to be best fit. An Anderson–Darling test statistic and p-value were computed as 0.72 & 0.54 respectively. The p-value is above the 0.05 significance interval.

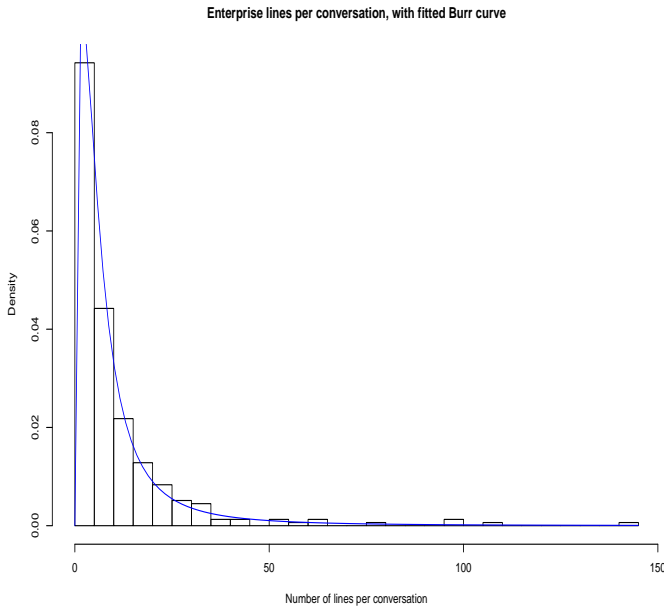


Fig. 11. Enterprise messages per conversation with fitted Burr curve

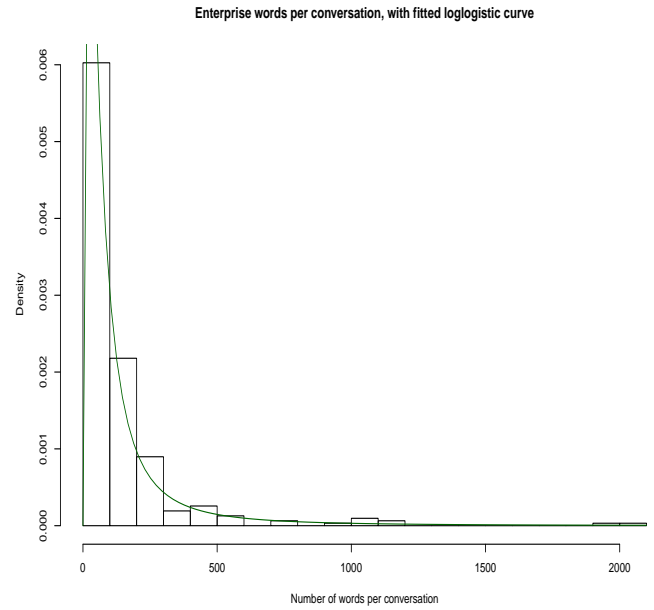


Fig. 12. Enterprise words per conversation with fitted loglogistic curve

#### D. Conversation messages & word modelling

Fig. 11 and Fig. 12 show probability density function histograms of both the messages and words per conversation for the enterprise dataset. A Burr distribution was found to be the best fit for messages per conversation. A loglogistic distribution was determined to be the best fit for words per conversation. An Anderson–Darling test statistic and p-value was computed for both distributions as 2.13 & 0.08 (messages per conversation dataset) and 0.65 & 0.6 (words per conversation dataset). In both cases the p-value was found to be above the 0.05 significance interval.

Fig. 13 and Fig. 14 show probability density function histograms of both the messages and words per conversation for the Ubuntu dataset. For both datasets, a Burr distribution was found to be the best fit. An Anderson–Darling test statistic and p-value was computed for both distributions as 1.76 & 0.13 (messages per conversation dataset) and 0.19 & 0.99 (words per conversation dataset). In both cases the p-value was found to be above the 0.05 significance interval.

#### E. Conversation user count modelling

Fig 15 illustrates the probability density function (PDF) and cumulative density function (CDF) plots of user counts per conversation for the enterprise dataset. However using the raw counts a Poisson distribution was found to be a poor fit due to an under-dispersion within the dataset. A hurdle method was implemented whereby the counts of  $n-1$  users were modelled. A chi-squared test statistic, degrees of freedom and p-value were calculated with the hurdle method applied. The values computed were 2.97, 4 and 0.56 respectively. It was noted the p-value was above the 0.05 significance.

Fig 16 illustrates the PDF and cumulative density function CDF plots of user counts per conversation for the enterprise

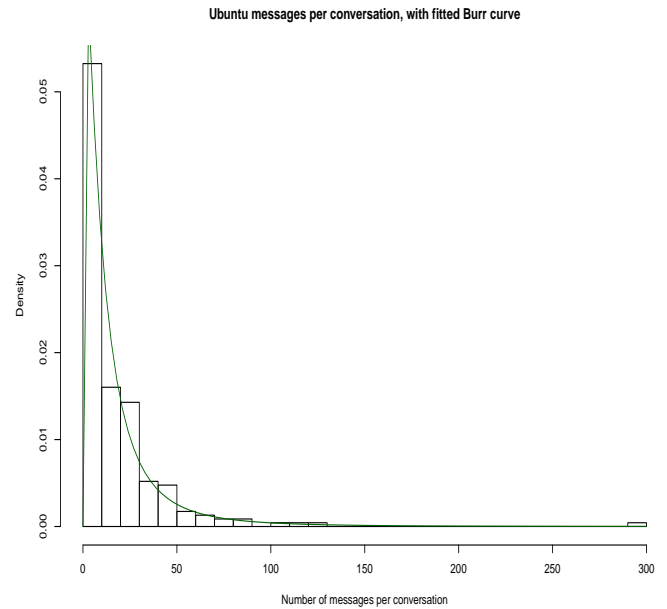


Fig. 13. Ubuntu messages per conversation with fitted Burr curve

dataset. However using the raw counts a Poisson distribution was found to be a poor fit. A similar hurdle method was applied to the count data as described for the enterprise data set. A chi-squared test statistic, degrees of freedom and p-value were calculated with the hurdle method applied. The values computed were 11.08, 5 and 0.05 respectively. It was noted the p-value was exactly 0.05 and not above the 0.05 significance.



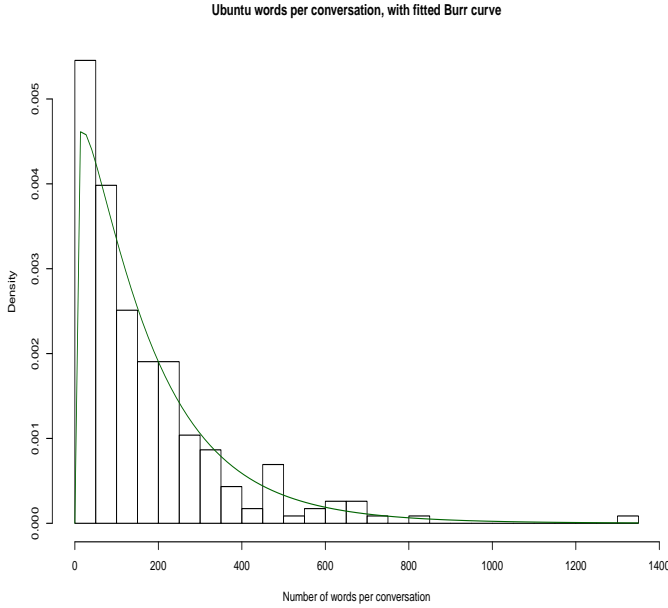


Fig. 14. Ubuntu words per conversation with fitted Burr curve

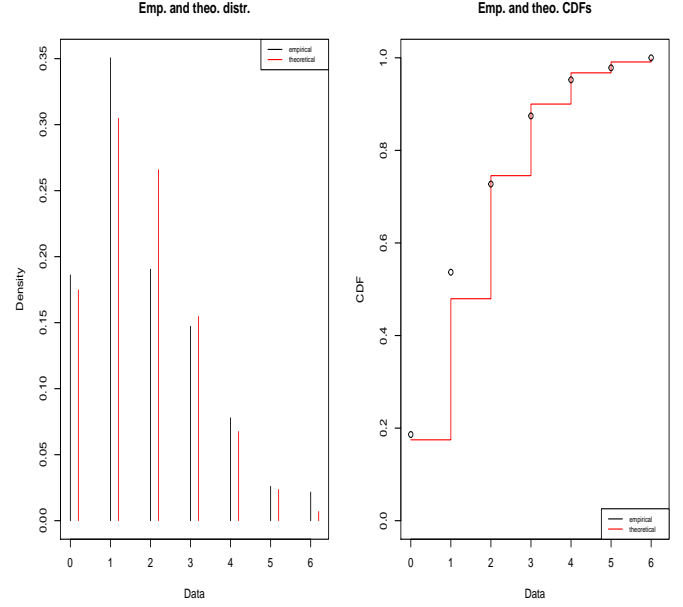


Fig. 16. Ubuntu N-1 users per conversation with fitted Poisson PDF and CDF

## V. DISCUSSION

Section IV provided a summary of modelling experiments that were conducted as part of his study. The following section provides deeper analysis and discussion of the results. In each section references will be made to each research question asked in section III.

Prior to a detailed discussion of our results, we summarise the results along with each corresponding research question. Table II provides this summary.

### A. Conversation duration modelling

The results section has shown that a parametric approach to model conversation durations is reasonable. For the enterprise dataset the Weibull distribution was the best fit. For the Ubuntu data set either a Burr or log-logistic distribution proved to be the best fit. In both cases we remark that the p-value for the Anderson-Darling Goodness of fit was well above the 0.05 significance interval.

Of interest is that, in order to produce the above fit, given that conversation duration were measured in minutes by the system logs, any durations of 0 minutes were removed from the dataset. The removal of 0's from a data set is typically undertaken as part of a hurdle model technique. We feel this is a reasonable approach as we are primarily interested in modelling conversations of a positive duration. It should be noted that the percentage of conversations removed were 23% and 17% for the enterprise and Ubuntu data sets respectively.

This study as has answered our first research question: Can the duration of annotated chat conversations be modelled by a parametric method. Data analysts from micro teams and startups can use the result of this work to compute a mean and standard deviation for their modelled distribution. These

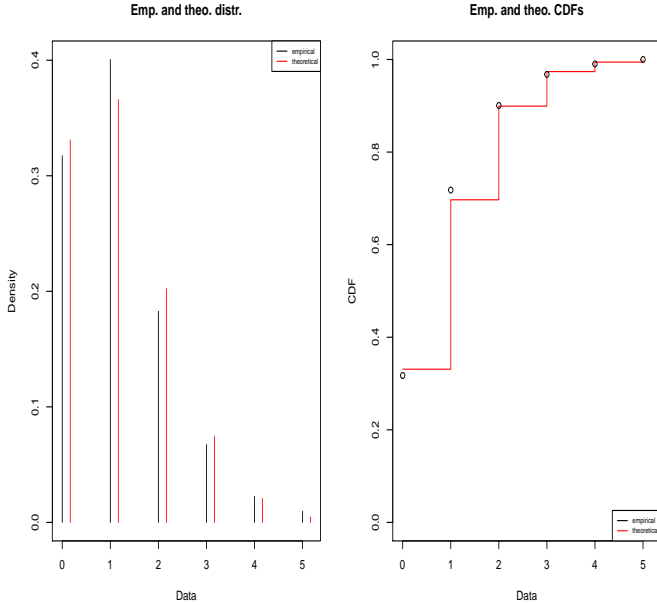


Fig. 15. Enterprise N-1 users per conversation with fitted Poisson PDF and CDF

TABLE II. SUMMARY OF RESEARCH QUESTION, RESULTS AND TECHNIQUES USED

Dataset	Research Question	Results	Data Transformation & Fitting techniques
Enterprise	1. What method can be used to model conversation duration times?	Weibull distribution is the best fit. AD test statistic = 1.30 p-value = 0.23	Zero values removed using Hurdle method. [24] MLE [14] [15]
Ubuntu	1. What method can be used to model conversation duration times?	Burr / loglogistical distribution is the best fit. AD test statistic = 1.10 p-value = 0.31	Zero values removed using Hurdle method. [24] MLE [14] [15]
Enterprise	2. What method can be used to model conversation delta times?	(Combined) No suitable parametric fit found. (Combined) Uniform kernel with SJ-dpi bandwidth = 56.73 (Entangled) Weibull distribution is the best fit. (Entangled) AD test statistic = 0.30, p-value = 0.94 (Logical) Weibull distribution is the best fit. (Logical) AD test statistic = 0.94, p-value = 0.76	(Combined) KDE Non-Parametric method used. [25] [26] [29] (Entangled) Absolute values used. MLE [14] [15] (Logical) MLE [14] [15]
Ubuntu	2. What method can be used to model conversation delta times?	(Combined) No suitable parametric fit found. (Combined) Gaussian kernel with rule-of-thumb bandwidth = 2.94 (Entangled) loglogistical distribution is the best fit. (Entangled) AD test statistic = 0.60, p-value = 0.64 (Logical) loglogistical distribution is the best fit. (Logical) AD test statistic = 2.46, p-value = 0.052	(Combined) KDE Non-Parametric method used. [25] [26] [28] (Entangled) Absolute values used. MLE [14] [15] (Logical) A 1 minute value was added to each delta time (x+1). MLE [14] [15]
Enterprise	3. What method can be used to model conversation inter-arrival times?	Weibull distribution is the best fit. AD test statistic = 0.78 p-value = 0.50	MLE [14] [15]
Ubuntu	3. What method can be used to model conversation inter-arrival times?	loglogistical distribution is the best fit. AD test statistic = 0.72 p-value = 0.54	A 1 minute value was added to each inter-arrival time (x+1). MLE [14] [15]
Enterprise	4. What method can be used to model conversation message and word counts?	(Messages) Burr distribution is the best fit. (Messages) AD test statistic = 2.13, p-value = 0.08 (Words) loglogistical distribution is the best fit. (Words) AD test statistic = 0.65, p-value = 0.60	MLE [14] [15]
Ubuntu	4. What method can be used to model conversation message and word counts?	(Messages) Burr distribution is the best fit. (Messages) AD test statistic = 1.76, p-value = 0.13 (Words) Burr distribution is the best fit. (Words) AD test statistic = 0.19, p-value = 0.99	MLE [14] [15]
Enterprise	5. Can a Poisson distribution be used to model conversation user counts?	Strong evidence to suggest Poisson is a good fit. $\chi^2 = 2.97$ degrees of freedom = 4 p-value = 0.56	User counts were reduced by 1 for all values and n-1 users were modelled and fitted. [24]
Ubuntu	5. Can a Poisson distribution be used to model conversation user counts?	Borderline evidence to suggest Poisson is a good fit. $\chi^2 = 11.08$ degrees of freedom = 5 p-value = 0.05	User counts were reduced by 1 for all values and n-1 users were modelled and fitted. [24]

measures of location can then be used to compute the expected duration of a conversation and the proportion of conversations that will last a fixed duration. If we think of conversations within a real-time messaging application as vehicle to discuss and diagnose complex problems, this result can be used as a way to model service time diagnosis and resolution.

Finally it should be noted that for each dataset, a different distribution result was produced. As we have noted previously the Ubuntu dataset has a greater ratio of messages per hour, it seems intuitive that a heavier tailed distribution (log-logistic) would be an appropriate fit.

### B. Conversation delta time modelling

We have learned from our results that no suitable parametric could be found to model overall conversation delta times. As we used a method to differentiate between entangled and logical delta times, a two tailed histogram was produced. We have seen that by using a non-parametric such as KDE, a suitable bandwidth selector and kernel shape can be computed. Once again we can see the results vary depending on the dataset used. For the enterprise data set a uniform kernel with a Sheather Jones direct plugin was found to be the most appropriate method and fit. For the Ubuntu data a Gaussian

kernel using Silverman's rule-of-thumb bandwidth selector yielded the best approach.

Conversely our study found that by dividing the conversation delta times into entangled and logical delta subsets, a parametric method can be used for data modelling. Initiatively we found that Weibull and log-logistic distributions were the most appropriating fitting distributions. We remark, that these distributions are the same as the ones used to model conversation duration. In all cases they p-value of each fit exceeded the 0.05 significance interval. However we note that for modelling of logical delta times from the Ubuntu data set, our p-value was computed to three decimal places to ensure the p-value was greater than the significance interval. This specific result should be treated with some caution.

This piece of research has answered our second research question. Clearly for datasets with entangled and logical delta times, a non-parametric approach is our preferred option. However if a parametric approach is required, by subsetting the data, a result (showing an appropriate distribution may be possible). If we think of the conversation delta times as the downtime between conversations, location measures can be computed. These measures can then be used to forecast the downtimes of team discussion in a collaborative environment.

### C. Conversation inter-arrival time modelling

For our third research question we asked what is the most appropriate method to model conversation inter-arrival times. We learned that once again the Weibull and log-logistic distributions were the most appropriate fits for the enterprise and Ubuntu datasets respectively. We know that the inter-arrival time is a function of conversation duration and delta times. Therefore it's intuitive that same type of distribution was found to model a time period which spans both the duration and delta times. For both data sets we note that the p-values for goodness of fit exceeded the 0.05 confidence interval.

Of interest, for the Ubuntu data set a small constant (1 minute) was added to each inter-arrival time duration. Upon review of the data a small number of inter-arrival times were found to be of 0 minute duration. Due most likely to dense bursts of messages in a collective conversation thread. Rather than remove these data points a small data transformation was applied. We note that this constant effects the overall scale of the dataset rather than the underlying shape.

The result from this work as helped answer our third research question. We wanted to understand whether inter-arrival times between conversations could be modelled effectively by a parametric method. As we can see that a parametric approach is plausible. Data scientists from startups and micro teams can use this result in two ways. As we have seen conversation duration, delta, and inter-arrival time all share a common data set on a per dataset basis, we believe this to be no coincidence given that inter-arrival time is a function of duration and delta time. Furthermore we believe that the inter-arrival time results combined with a service time result (conversation duration) could be used as part of a queuing framework to model conversation busy times on a daily basis.

### D. Conversation messages & word modelling

Our fourth research question moved the focus from conversation duration to modelling it's constituent parts: the words used and the number of messages in a conversation. We determined that, for the enterprise dataset a Burr (Messages) and log-logistic (Words) were the most appropriate fits. For the Ubuntu dataset a Burr distribution (Messages & Words) was the most appropriate fit. For all four distribution fitting results we note that the p-value exceeded the 0.05 confidence interval.

The Burr distribution is a flexible distribution that can illustrate a wide range of distribution shapes and types, due to the three parameters that are used to define it's shape. The Burr distribution was initially used in finance (to express income levels), but has grown to wider use in areas such as hydrology (flood level modelling) and reliability (failure rates of components).

While three of the four datasets were fitted with a Burr distribution we note that the enterprise words distribution was best fitted with a log-logistic distribution, given that the Burr distribution is sometimes referred to as a generalised log-logistic distribution we know there is a close affinity between both of these heavy tailed distribution, as such this result is not unsurprising.

TABLE III. UBUNTU: USER COUNTS PER CONVERSATION (UNTRANSFORMED HURDLE ADJUSTMENT)

Transformation	0 users	1 user	2 users	3 users	4 users	5 users	6 users	7 users
Untransformed	0	43	81	44	34	18	6	5
N-1 users	43	81	44	34	18	6	5	NA

We have shown that a parametric approach can provide suitable result in terms of fitting messages and words to a distribution. Additionally that a heavy-tailed distribution should be used as a first port of call for distribution fitting. Using this result micro teams and startups can model their conversations to determine the expected number of messages required to conduct a conversation. Additionally this result can be used to aid future work in the area of topic analysis of chat conversations. By understanding the expected message and words counts, suitable topic clusters and top term values can be seeded from word and message distributions.

### E. Conversation user count modelling

Our final research question centred around whether a suitable method can be used to model the counts of users who participate in a group chat conversation. Typically for count data, a first choice is to fit a Poisson distribution. We learned that fitting a Poisson distribution to the untransformed user count data was not a good fit for either dataset. Upon more in-depth analysis we checked to determine the level of under / over dispersion within both each data set. It was noted that in both datasets, evidence of under dispersion was found.

In order to correct the under-dispersion a hurdle method was adopted to mitigate. Rather than remove the 0 count bin from the data set, we reduced the bin count by 1 for each dataset. Table III illustrates the Ubuntu user counts per conversation before and after the hurdle adjustment. We found that with the hurdle adjusted data set the Poisson distribution was a good fit, for the enterprise dataset with a p-value well in excess of the 0.05 confidence interval. However we found that the Poisson distribution was a borderline fit for the Ubuntu data with a p-value of exactly 0.05. We remark that with the hurdle adjustment, the under dispersion was greatly reduced within the enterprise dataset however for the Ubuntu data set some under dispersion remained after the hurdle adjustment was applied.

While the counts of users could not be modelled directly, we feel modelling of n-1 users with a Poisson distribution is a valuable result. Micro teams and startups can use this result to further future research into the field of conversation analysis. By combining conversation topic and user count modelling an enhanced model could be derived. This model could help infer what conversation topics attract large numbers of users and for large group conversations, can active and passive subsets be identified?

## VI. CONCLUSION

The purpose of this study was to determine what parametric / non-parametric fitting techniques could be used to model data generated from real-time chat conversations using two separate data sets. We found that a "one size fits all approach" is not appropriate, rather a combination of approaches are

required to adequately fit such data. The findings of this study support previous study specifically in the field of internet chat discourse inter-arrival time modelling.

Previous studies have shown that the inter-arrival times of internet chat messages can be classified as heavy-tailed datasets. We have shown that when using a parametric approach, such times can be modelled by a log-logistic or Weibull distributions.

This work provides a more comprehensive study, specifically in relation to modelling multiple facets of real-time chat conversations (i.e. Chat duration, delta, inter-arrival and users per chat), and clearly illustrates that depending on the dataset the results are different every time.

In future micro teams and startups can assess their chat conversation data to understand how conversation times are structured within their teams. A specific chat conversation analysis framework can be developed to allow teams to surface inter-arrival time and service time of problem determination resolution.

In subsequent work we shall target more deeply the inter-arrival time of both chat chat conversations and single messages. Our initial focus will determine whether the burst times of conversations and messages can be modelled by a Markov process.

#### ACKNOWLEDGMENT

The authors would like to thank William Huber for his helpful suggestion for fitting under dispersed count data to a Poisson distribution [45].

#### REFERENCES

- [1] M. Pendolino. (2017) 3 ways collaborative software can solve enterprise challenges. [Online]. Available: <http://bit.ly/2uzU480>
- [2] K. Wolf. (2017) 8 business problems the best collaboration software can solve. [Online]. Available: <http://bit.ly/2vfreHX>
- [3] M. Haughey. (2017) Setting up slack for small teams. [Online]. Available: <http://bit.ly/2vzw81Q>
- [4] "Watson workplace," 2017. [Online]. Available: <https://ibm.co/2uG4ZgW>
- [5] "Slack," 2017. [Online]. Available: <http://bit.ly/1uEVWVc>
- [6] "Microsoft teams," 2017. [Online]. Available: <http://bit.ly/2tngfPb>
- [7] "Azendoo," 2017. [Online]. Available: <http://bit.ly/1lnHcX5>
- [8] C. A. Sottile, "Sick of email? slack wants to kill your inbox clutter," 2017. [Online]. Available: <http://nbcnews.to/2uyzLYR>
- [9] C. Fowler, "How to avoid email paralysis," 2017. [Online]. Available: <http://bit.ly/2vhWIT1>
- [10] "How the engineering team at ibm uses slack throughout the development lifecycle," 2017. [Online]. Available: <http://bit.ly/2qcd07G>
- [11] C. Boulton, "How devops, agile spurred slack enterprise adoption," 2017. [Online]. Available: <http://bit.ly/2kTMHB8>
- [12] L. Kleinrock, *Queueing systems*. Wiley, 1975.
- [13] K. Pearson, "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894.
- [14] R. A. Fisher, "Theory of statistical estimation," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 22, no. 5. Cambridge University Press, 1925, pp. 700–725.
- [15] S. S. Wilks, "The large-sample distribution of the likelihood ratio for testing composite hypotheses," *The Annals of Mathematical Statistics*, vol. 9, no. 1, pp. 60–62, 1938.
- [16] H. Cramér, *On the composition of elementary errors*. Almqvist & Wiksells, 1928.
- [17] R. Von Mises, "Statistik und wahrheit," *Julius Springer*, 1928.
- [18] N. Smirnov, "Table for estimating the goodness of fit of empirical distributions," *The annals of mathematical statistics*, vol. 19, no. 2, pp. 279–281, 1948.
- [19] T. W. Anderson and D. A. Darling, "A test of goodness of fit," *Journal of the American statistical association*, vol. 49, no. 268, pp. 765–769, 1954.
- [20] —, "Asymptotic theory of certain" goodness of fit" criteria based on stochastic processes," *The annals of mathematical statistics*, pp. 193–212, 1952.
- [21] B. M. Hill *et al.*, "A simple general approach to inference about the tail of a distribution," *The annals of statistics*, vol. 3, no. 5, pp. 1163–1174, 1975.
- [22] J. Pickands III, "Statistical inference using extreme order statistics," *the Annals of Statistics*, pp. 119–131, 1975.
- [23] J. Nair, A. Wierman, and B. Zwart, "The fundamentals of heavy-tails: properties, emergence, and identification," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1. ACM, 2013, pp. 387–388.
- [24] J. Mullahy, "Specification and testing of some modified count data models," *Journal of econometrics*, vol. 33, no. 3, pp. 341–365, 1986.
- [25] M. Rosenblatt *et al.*, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.
- [26] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [27] V. A. Epanechnikov, "Non-parametric estimation of a multivariate probability density," *Theory of Probability & Its Applications*, vol. 14, no. 1, pp. 153–158, 1969.
- [28] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.
- [29] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 683–690, 1991.
- [30] C. Dewes, A. Wichmann, and A. Feldmann, "An analysis of internet chat systems," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*. ACM, 2003, pp. 51–64.
- [31] M. Lukasik, P. Sriji, T. Cohn, and K. Bontcheva, "Modeling tweet arrival times using log-gaussian cox processes," in *EMNLP*, 2015, pp. 250–255.
- [32] C. Vande Kerckhove, B. Gerencsér, J. M. Hendrickx, and V. D. Blondel, "Markov modeling of online inter-arrival times," *arXiv preprint arXiv:1509.04857*, 2015.
- [33] N. M. Markovitch and U. R. Krieger, "Nonparametric estimation of long-tailed density functions and its application to the analysis of world wide web traffic," *Performance Evaluation*, vol. 42, no. 2, pp. 205–222, 2000.
- [34] R. E. Maiboroda and N. M. Markovich, "Estimation of heavy-tailed probability density function with application to web data," *Computational Statistics*, vol. 19, no. 4, p. 569, 2004.
- [35] (2017) Visualizing inter-arrival times of tweets. [Online]. Available: <http://bit.ly/2tm8AMeHP>
- [36] (2017) Ubuntu irc logs. [Online]. Available: <https://irclogs.ubuntu.com/>
- [37] M. Elsner and E. Charniak, "Disentangling chat," *Computational Linguistics*, vol. 36, no. 3, pp. 389–409, 2010.
- [38] M. L. Delignette-Muller and C. Dutang, "fitdistrplus: An R package for fitting distributions," *Journal of Statistical Software*, vol. 64, no. 4, pp. 1–34, 2015. [Online]. Available: <http://www.jstatsoft.org/v64/i04/>
- [39] C. J. G. Bellosta. R package adgoftest. [Online]. Available: <http://bit.ly/1NU3c5y>
- [40] R package density. [Online]. Available: <http://bit.ly/2tm8AMe>
- [41] M. et al. R package vcd. [Online]. Available: <http://bit.ly/2vyTULd>
- [42] A. Z. Christian Kleiber. R package aer. [Online]. Available: <http://bit.ly/1LZy9pI>

[43] W. A. Huber. What to do when count data does not fit a poisson distribution. [Online]. Available: <http://bit.ly/2uHCkIx>