Predicting the Severity of Support Tickets from Machine Event Data

Shriya Mittapalli
Masters in Data Science, FAU
Machine Learning and Data Analytics for Industry 4.0 Seminar
Erlangen, Germany
shriya.mittapalli@fau.de

***Abstract*—Large volumes of customer requests are a common problem for modern service providers, who must respond to them quickly and efficiently to guarantee that customers receive the support they need. One scenario is to determine whether the ticket resolution can be provided remotely, or an onsite intervention is required. Many machine learning classification algorithms have been proposed to tackle this issue. Misclassification not only incurs financial losses for companies but also wastes valuable resources and time for both companies and customers. Historically it has been shown that the point estimation is unreliable in this context. One solution is to measure the uncertainty of predictions, which provides insights into the reliability of classifications, enabling better decision-making.**

In this study, two methods for uncertainty estimation are proposed: the Probability Range and Standard Deviation approaches. These approaches are discussed within the context of the proposed system, which comprises Monte Carlo Dropout and Ensemble models. The probability Range can be used to measure the model uncertainty as a whole, whereas the Standard Deviation approach can be used to measure the uncertainty of each prediction, along with the overall model uncertainty. It is demonstrated that the two proposed approaches can serve as reliable measures of uncertainty.

***Keywords*—Uncertainty Estimation, Monte Carlo Dropout, Ensemble Model, Sequential Dense Model, Gated Recurrent Unit (GRU), Probability Range, Standard Deviation**

I. Introduction

The problem of support ticket classification has a huge impact on companies. A customer's request for assistance from a service provider's support team is referred to as a support ticket. Tickets serve as the most important means of communication between users and the staff responsible for the management of a service, facilitating the resolution of any issue or incident related to it. The common examples include IT-related support requests, bug reports (Mani et al. [1]), healthcare (Young et al. [2]), and governmental institutions (Powell et al. [3]).

When a ticket is generated, it is crucial to classify it and assign it to a resolving specialist. Tickets must be resolved within a set amount of time to guarantee customer satisfaction and high productivity (Gupta et al. [4]). Customer service staff are often confronted with prioritizing reported machine problems (tickets) according to their severity. One important distinction is whether a problem can be solved remotely or if a customer service engineer must travel to the customer site. This involves considering different factors and is tedious. Any mistakes in this classification cause delays in ticket resolution and unnecessary onsite visits which in turn causes improper resource utilization and customer dissatisfaction.

With the substantial increase in the number of support tickets (Ali Zaidiet al. [5]; Fuchs et al. [6]), there is an increasing demand for automated systems that can accelerate the ticket resolution process. Hence, there is a need to develop a model that classifies the tickets into remote or on-site resolution. Unreliable prediction causes delays, unnecessary reassignment, and sub-optimal resource utilization, all of which have a negative financial impact on service providers as well as clients. Understanding the confidence level of a model's prediction is crucial.

The main aim of this work is to research solutions for uncertainty estimations in Deep Neural Networks and apply them to the problem of predicting the severity of support tickets and also provide insights about the reliability of predictions.

II. Related Work

In the context of ticket classification, the majority of research papers either suggest novel approaches or examine how they operate inside a certain domain. A few reviews and surveys address this topic. For instance, Revina et al. [7] examine text representation methods and the efficacy of different text classifiers when discussing ticket classification in the IT area. They do, however, restrict their review to classic classification techniques like Random Forests and Support Vector Machines. They explore the importance of explainable ticket classification and the variables that affect prediction quality.

There are number of research which identifies the reliability of model prediction. Giovanna Nicora et al. [8] review approaches to identify unreliable predictions and highlight overlap among concepts. This is achieved by implementing the density principle and the local fit principle. The similarity between the instance to be evaluated and the training set is confirmed by the density principle. The trained model's effectiveness is confirmed by the local fit principle on training subsets that are more like the instance being assessed.

Peter Schulam et al. [9] describes Resampling Uncertainty Estimation (RUE), an algorithm to audit the point-wise reliability of predictions. Intuitively, RUE estimates the amount that a prediction would change if the model had been fit on different training data. Jakob Gawlikowski et al. [10] delves deep into uncertainty estimation in Deep Neural Networks and suggest different ways for uncertainty estimation.

It has been shown unreliable in the past to predict the severity based on machine data with point estimations. One possible solution might be to provide uncertainty estimations for each model prediction, additionally to point estimations.

III. DATASET

The proposed model uses the Computed Tomography (CT) dataset, which was collected by Siemens Healthineers. The data is given in two different forms. The first one has information about all the tickets. Case_Id is a unique identifier for each ticket. Machine_Id represents each machine. Onsite is the label where 0 refers to the ticket that can be resolved remotely, and 1 refers to the ticket that has to be resolved onsite. Start_Time and End_Time are the times at which tickets have been created and closed respectively. This contains 7406 tickets of which 3696 belongs to onsite 0 and 3710 to onsite 1.

The second type has information about events and there are 116 such files. The events are collected from approximately the past 20 days up to the time of ticket creation. For each Case_Id, the sequence of events and its timestamps are provided.

Fig. 1 illustrates the steps for creating the dataset. 1635 Case_Ids do not have sequence of events, hence removed. Later identified the Events and Case_Ids to be removed. Events that appeared in less than 20 Case_Ids and those that appeared in more than 2500 Case_Ids were removed (285 events). Case_Ids with less than 25 unique events and greater than 150 unique events were discarded (10 Case_Ids). For each Case_Id, selected the last 8000 events or discarded otherwise. The final dataset contains 4321 data points.

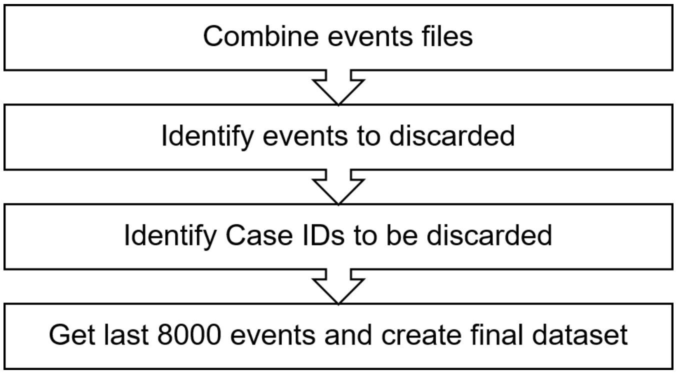

Combine events files
Identify events to discarded
Identify Case IDs to be discarded
Get last 8000 events and create final dataset

Fig. 1: Steps for Data Preprocessing

IV. METHODOLOGY

Two models have been presented for ticket classification with uncertainty estimation: Monte Carlo Dropout and Ensemble models. After preprocessing the dataset, the models are created. Two evaluation approaches are proposed and discussed along with their potential utility for uncertainty estimation.

A. Model Creation

Dropouts are used in Neural Networks during training to reduce overfitting. However, Monte Carlo Dropout goes beyond the traditional use of dropout in training and extends it to the inference phase. By using dropouts during inference, Monte Carlo Dropout produces multiple predictions for a single input, resulting in a more accurate measure of uncertainty in the model's predictions. The model summary is shown in Fig. 2. It is a sequential model with multiple dense layers. Flatten layer is used to map the input. There are 10 dense layers in total and each one is associated with a dropout except the output layer. Initially, there are 8192 neurons which gradually reduces to 1 neuron in the last layer. All the layers use the ReLU activation function except the output layer which uses the softmax activation function. There are about 1.1 Billion trainable parameters.

The Ensemble model uses multiple models and combines their results for reliable estimations. The model summaries for both Dense and GRU models are shown in Fig. 3. The Ensemble model includes four Dense models with different initial weights and a GRU model. The different weight initializers are Glorot Normal, He Normal, Orthogonal, and Variance Scaling. The Dense model is similar to the Monte Carlo Dropout model except the dropout is present only at the last layer. The GRU model has two GRU layers with ReLU activation. The output layer is a dense layer with 1 neuron and softmax activation which provides the prediction probability.

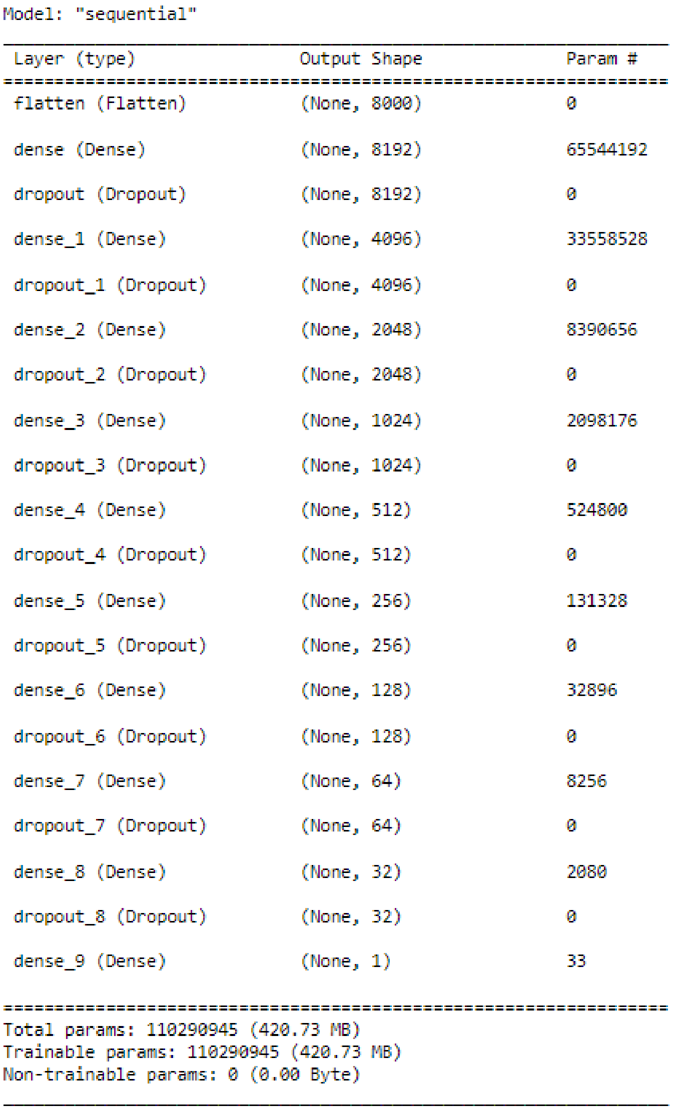
B. Model Training

Before training, the data is split into 3 parts. 60% of the data is used for training. 20% is used each for validation and testing. Same Machine_Id is not present in multiple partitions. 60% of cases with onsite label as 1 and 60% of cases with onsite label as 0 are taken for train data. The remaining data is divided equally into test and validation data keeping the class balance.

During training, Adam optimizer is used for weights update, and Binary Cross Entropy is used as a loss function as this is a binary classification problem. The model is trained for 1000 epochs with early stopping. The early stopping monitors the validation loss. Training is stopped if the change in validation loss is less than 0.001 for 5 consecutive epochs. For the Ensemble model, the training data is shuffled for each model, along with different weight initialization to cause variations in models.

C. Model Evaluation

For Monte Carlo Dropout, each sample is measured 20 times in training mode. When this is done, different dropouts are applied each time to obtain different output probabilities. Similarly for the Ensemble model, instead of 20 predictions, 5 model predictions are used. Two approaches are proposed for



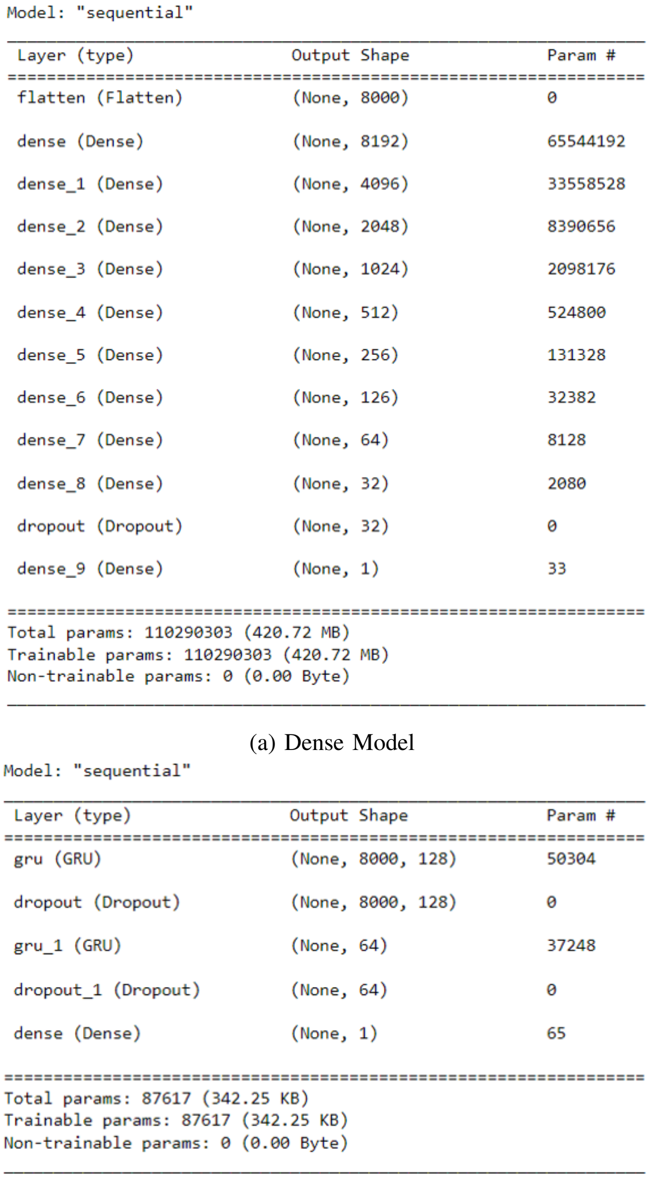
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 8000)              0

 dense (Dense)               (None, 8192)              65544192

 dropout (Dropout)           (None, 8192)              0

 dense_1 (Dense)             (None, 4096)              33558528

 dropout_1 (Dropout)         (None, 4096)              0

 dense_2 (Dense)             (None, 2048)              8390656

 dropout_2 (Dropout)         (None, 2048)              0

 dense_3 (Dense)             (None, 1024)              2098176

 dropout_3 (Dropout)         (None, 1024)              0

 dense_4 (Dense)             (None, 512)               524800

 dropout_4 (Dropout)         (None, 512)               0

 dense_5 (Dense)             (None, 256)               131328

 dropout_5 (Dropout)         (None, 256)               0

 dense_6 (Dense)             (None, 128)               32896

 dropout_6 (Dropout)         (None, 128)               0

 dense_7 (Dense)             (None, 64)                8256

 dropout_7 (Dropout)         (None, 64)                0

 dense_8 (Dense)             (None, 32)                2080

 dropout_8 (Dropout)         (None, 32)                0

 dense_9 (Dense)             (None, 1)                 33

=================================================================
Total params: 110290945 (420.73 MB)
Trainable params: 110290945 (420.73 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 2: Model Summary for Monte Carlo Dropout

uncertainty estimation namely, Probability Range and Standard Deviation.

In the Probability Range approach, the average probability of the predictions is taken and plotted for all samples to identify the threshold boundaries. Accuracy is calculated only for the points which satisfy these thresholds. The data points which fail these conditions are used as a measure of uncertainty.

In the Standard Deviation approach, the probability average of predictions is considered along with the standard deviation. A high standard deviation indicates high uncertainty. If the average prediction probability for a particular data point, plus or minus the standard deviation, belongs to the same class then the prediction is assigned to that class. Otherwise, it is considered uncertain.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 8000)              0

 dense (Dense)               (None, 8192)              65544192

 dense_1 (Dense)             (None, 4096)              33558528

 dense_2 (Dense)             (None, 2048)              8390656

 dense_3 (Dense)             (None, 1024)              2098176

 dense_4 (Dense)             (None, 512)               524800

 dense_5 (Dense)             (None, 256)               131328

 dense_6 (Dense)             (None, 126)               32382

 dense_7 (Dense)             (None, 64)                8128

 dense_8 (Dense)             (None, 32)                2080

 dropout (Dropout)           (None, 32)                0

 dense_9 (Dense)             (None, 1)                 33

=================================================================
Total params: 110290303 (420.72 MB)
Trainable params: 110290303 (420.72 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

(a) Dense Model

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 8000, 128)         50304

 dropout (Dropout)           (None, 8000, 128)         0

 gru_1 (GRU)                 (None, 64)                37248

 dropout_1 (Dropout)         (None, 64)                0

 dense (Dense)               (None, 1)                 65

=================================================================
Total params: 87617 (342.25 KB)
Trainable params: 87617 (342.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

(b) Gated Recurrent Unit Model

Fig. 3: Model Summary for Ensemble Model

The uncertainty and accuracy is measured by,

$$Uncertainty = \frac{\#UncertainPredictions}{\#TotalSamples} \tag{1}$$

$$Accuracy = \frac{\#CorrectPredictions}{\#TotalPredictions} \tag{2}$$

where:

$CorrectPredictions$ = number of samples model predicted accurately
$UncertainPredictions$ = number of samples not classified by model
$TotalSamples$ = total number of samples
$TotalPredictions$ = Total Samples - Uncertain Predictions

TotalSamples – UncertainPredictions gives the total no of samples that the model assigns a class to. Accuracy is calculated as the number of data points that the model assigns the correct class out of the total number of data samples that the model assigns to a particular class. Uncertainty is the number of samples that the model failed to assign class.

V. Results and Discussion

According to the Probability Range approach (Table I), both the models are about 75% and 90% uncertain and about 55% accurate. For the Standard Deviation approach (Table II), the Monte Carlo Dropout model is highly uncertain as the standard deviation is high and hence it has an accuracy of about 50%. The standard deviation is comparatively less for the Ensemble model, but still, it shows a high uncertainty of about 60% with an accuracy of around 52%. This looks like the standard deviation may not be a good measure of uncertainty for the Ensemble model. This is because the Ensemble model uses four similar sequential models with different initial weights which do not produce much variations in probability. If different models were used, then standard deviation could be a reliable measure for uncertainty estimation.

Model	Measure	Validation Data	Test Data
Monte Carlo Dropout	Uncertainty	74.53%	74.45%
	Accuracy	50.45%	55.66%
Ensemble	Uncertainty	90.04%	89.31%
	Accuracy	59.14%	56.44%

TABLE I: Probability Range Approach

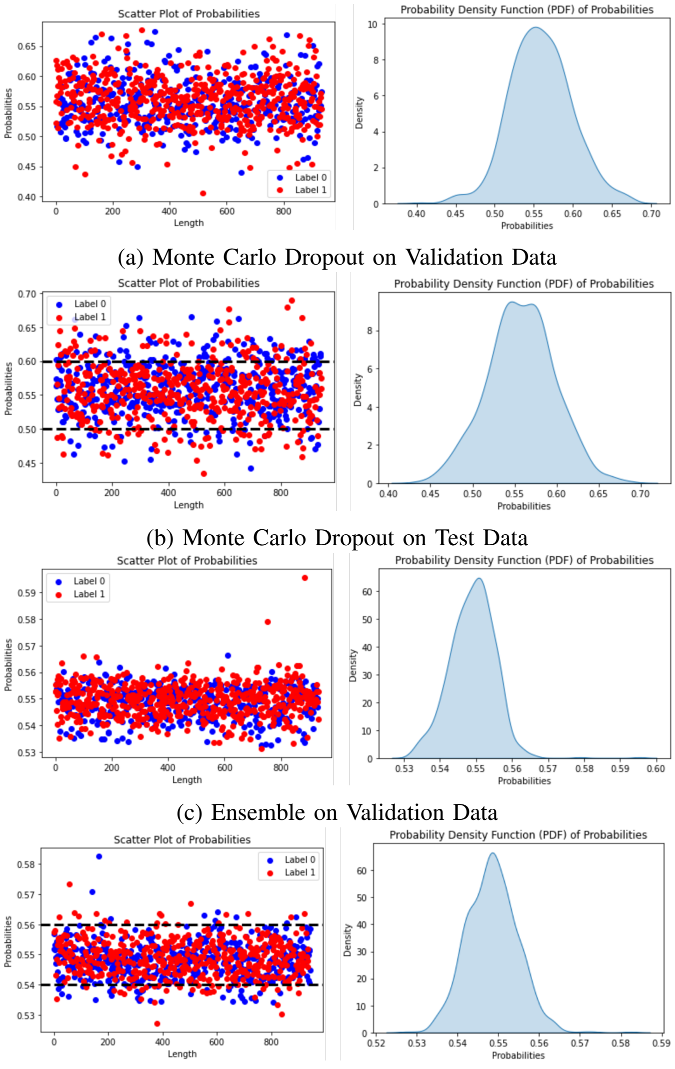
Model	Measure	Validation Data	Test Data	SD (σ)
Monte Carlo Dropout	Uncertainty	93.05%	92.02%	0.13
	Accuracy	50.00%	50.72%	
Ensemble	Uncertainty	57.67%	57.67%	0.0104
	Accuracy	50.00%	52.50%	

TABLE II: Standard Deviation Approach

Fig. 4 (a) and (b) shows the scatter plot and probability distribution of predictions for validation and test datasets respectively for the Monte Carlo Dropout model. There is no separation in probabilities between 0.5 and 0.6 for the validation data. These values are chosen as thresholds and applied on test data, demonstrating consistent performance. Similarly, Fig. 4 (c) and (d) shows the scatter plot and probability distribution of predictions for validation and test datasets respectively for the Ensemble model. The threshold values are 0.54 and 0.56.

The standard deviation is calculated for the entire model. A high standard deviation indicates the overall model is uncertain. Similarly, standard deviation can be computed for each prediction, providing insight into the uncertainty associated with individual predictions.

Despite the models exhibiting poor performance due to lack of information on events, the performance consistency between the validation and test datasets is preserved. This suggests that the two approaches proposed are viable uncertainty measures.



(a) Monte Carlo Dropout on Validation Data

(b) Monte Carlo Dropout on Test Data

(c) Ensemble on Validation Data

(d) Ensemble on Test Data

Fig. 4: Scatter Plots and Probability Distributions

VI. Conclusion and Future Work

In this work, two models are developed to classify tickets into onsite or remote resolution along with uncertainty measures to differentiate reliable and unreliable estimations. Standard deviation can be used as a measure of uncertainty. For the Ensemble model, use different varieties of models to get a better evaluation. The Probability Range approach can be used to measure the overall uncertainty of the model but not on individual predictions.

Despite the valuable insights gained, the study has certain limitations. The model's performance is suboptimal and requires improvement. Currently, there is no information about events and hence cannot perform informed decisions during data processing and model creation. Furthermore, model performance can be improved using complex models like Transformer models and Long Short-Term Memory with Gaussian Mixture Models.

References

[1] Senthil Mani, Anush Sankaran, and Rahul Aralikatte, '"DeepTriage: Exploring the Effectiveness of Deep Learning for Bug Triaging", In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (CODS-COMAD '19). Association for Computing Machinery, New York, NY, USA, pp. 171–179, Jan. 2019.

[2] Young IJB, Luz S, Lone N, "A systematic review of natural language processing for classification tasks in the field of incident reporting and adverse event analysis", Int J Med Inform, vol. 132(103971), Oct. 2019

[3] M. Powell, J. A. Rotz, and K. D. O'Malley, "How Machine Learning Is Improving U.S. Navy Customer Support", AAAI, vol. 34, no. 08, pp. 13188–13195, Apr. 2020.

[4] Gupta, H.S., Sengupta, B., "Scheduling Service Tickets in Shared Delivery". In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds) Service-Oriented Computing. ICSOC 2012. Lecture Notes in Computer Science, vol 7636. Springer, Berlin, Heidelberg, 2012.

[5] Zaidi, Syed and Fraz, Muhammad and Shahzad, Muhammad and Khan, Sharifullah, "A multiapproach generalized framework for automated solution suggestion of support tickets", International Journal of Intelligent Systems, vol. 37(6), pp. 3654–3681, Jun. 2022.

[6] Fuchs, Simon, and Drieschner, Clemens and Wittges, Holger, "Improving Support Ticket Systems Using Machine Learning: A Literature Review", In Proceedings of the 55th Hawaii International Conference on System Sciences, pp. 1893–1902, 2022.

[7] Revina, Aleksandra and Buza, Krisztian, and Meister, Vera G, "IT ticket classification: The simpler, the better", IEEE Access, Vol. 8, pp. 193380–193395, 2020.

[8] Giovanna Nicora, Miguel Rios, Ameen Abu-Hanna, Riccardo Bellazzi, "Evaluating pointwise reliability of machine learning prediction", Journal of Biomedical Informatics, vol. 127, Jan. 2022.

[9] Peter F. Schulam and Suchi Saria,"Can You Trust This Prediction? Auditing Pointwise Reliability After Learning", Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, vol. 89, 2019.

[10] Gawlikowski, Jakob and Tassi, Cedrique and Ali, Mohsin and Lee, Jongseok and Humt, Matthias and Feng, Jianxiang and Kruspe, Anna and Triebel, Rudolph and Jung, Peter and Roscher, Ribana and Shahzad, Muhammad and Yang, Wen and Bamler, Richard and Zhu, Xiao, "A survey of uncertainty in deep neural networks", Artificial Intelligence Review, vol. 56, pp. 1–77, 2023.