Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis Econometrics & Management Science: Business Analytics & Quantitative Marketing

Addressing Class Imbalance in Multi-Touch Conversion Attribution: A Realistic Approach

Pim Weterman (475771)

Supervisor:	dr. Kathrin Gruber
Second assessor:	dr. Flavius Frasincar
Date final version:	2nd of July, 2023

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Abstract

In online advertising, an internet user is continuously exposed to different advertisement campaigns. Accurately estimating contributions of advertisements to conversion decisions is crucial for marketers and advertisement auctioneers. However, the literature lacks simple and interpretable models that solve the multi-touch attribution (MTA) problem. In this paper, we discuss a simple, interpretable, feed-forward neural network using attention mechanisms, which we refer to as a Stacked Web of Attentional Neurons (SWAN). We show that the SWAN performs similar as a state-of-the-art model and adds interpretability, despite its simpler architecture. On top of that, we introduce a novel ensemble learning approach, referred to as Ensemble-SWAN, which uses random undersampling to leverage the class imbalance problem in MTA data. The Ensemble-SWAN shows slightly reduced performance compared to the SWAN. However, we believe using the Ensemble-SWAN is worthwhile as failures in real-world applications caused by oversampling are avoided. On top of that, to address the need for acknowledging uncertainty in real-world applications, we discuss two epistemic uncertainty quantification (UQ) approaches. In the constituent models of the Ensemble-SWAN, we observe notable presence of epistemic uncertainty, emphasizing the need for caution in real-world applications.

Acknowledgments

I would like to express my deepest gratitude to dr. Kathrin Gruber for her invaluable support and guidance throughout the entire duration of my master thesis. Her expertise and commitment have been instrumental in shaping the successful outcome of this research. On top of that, I thoroughly enjoyed the collaboration with dr. Gruber as her enthusiasm made the research process more engaging and fulfilling. Thank you so much!

Contents

1 Introduction

The advertising market has experienced an enormous shift from offline to online in the past decade. Newspaper advertisements, leaflets, billboards, radio advertisements, and television commercials have made way for advertisements on social media, Google, and via e-mail. Companies are increasingly relying on the online customer journey to persuade potential customers to buy their products. Moreover, the retail landscape is evolving rapidly into an omni-channel world, with new channels, media, and different types of devices competing for marketing investments (De Haan, Kannan, Verhoef & Wiesel, 2015). As enormous sums of money are spent on marketing, the careful allocation of marketing budget is one of the most crucial decisions a company has to make. In today's competitive business landscape, companies face constant pressure from stakeholders to maximize advertising returns, underscoring the critical importance of effective marketing campaigns.

The online customer journey of an internet-user often consists of multiple exposures to advertisements, called 'touchpoints' or 'impressions' and will eventually lead to a final purchase or not, called the conversion. An example of such an online customer journey is illustrated in Figure 1. Advertisers and advertisement exchanges can leverage browsing history and advertisement interaction to uncover meaningful insights on the effect of certain advertisements on potential conversion. As most online advertisements are sold by advertisement exchanges using real-time bidding (Cai et al., 2017), it is of critical importance for both advertisement sellers and marketers to accurately determine the value of certain touchpoints in the online customer journey. However, it remains a challenge to accurately attribute 'credit' to a certain touchpoint in a customer journey. Attributing credit to conversion in such an online user sequence is referred to as the multi-touch attribution (MTA) problem.

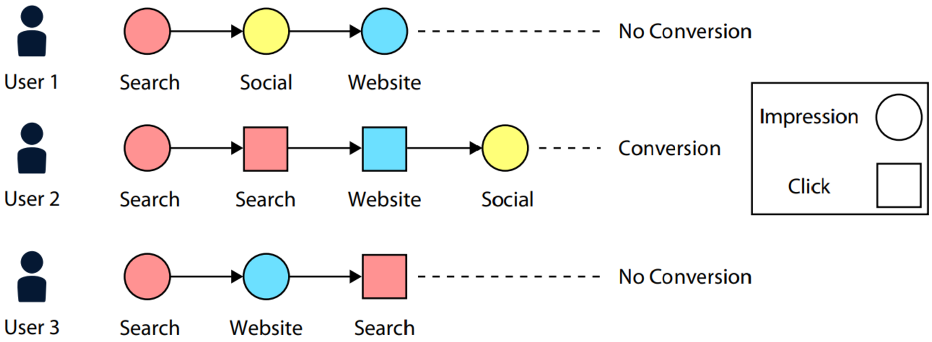


Figure 1: An illustration of different online user journeys consisting of advertisement impressions and clicks.

In the past decade, attribution modelling has transitioned from reliance on human intuition to reliance on more advanced mathematical models. Many different approaches to solve MTA problems are present in the literature. Initially, advertisers employed rule-based approaches such as the first-touch, last-touch, linear-touch, and time-decaying models to address MTA problems (Zhang, Wei & Ren, 2014; Wooff & Anderson, 2015; Buhalis & Volchek, 2021). These rule-based approaches were popular in practice due to their simplicity. On the other hand, empirical evidence and research has demonstrated inherent weaknesses in rule-based approaches used to address the MTA problem. For example, in Goldfarb and Tucker (2011), the authors find that obtrusive advertisements (e.g., intrusive, disruptive, overly prominent in presentation) have a negative relationship with conversion probability. Rule-based approaches are not able to capture these kind of complex dynamics.

To overcome the limitations mentioned above, data-driven approaches emerged as a solution. Amongst others, algorithmic approaches (Zhou et al., 2019; Dalessandro, Perlich, Stitelman & Provost, 2012; Xu et al., 2016), and approaches based on additive survival analysis (Zhang et al., 2014; Ji, Wang & Zhang, 2016) have been widely discussed in the literature. Due to the enormous growth in the use of mobile technological devices, the customer journey shifted to an omni-channel experience. This significantly increased the size and complexity of the to-be-solved MTA problems. As a result of this increase in dimensionality, deep learning approaches arose to tackle these complex and computationally heavy MTA problems. On top of that, deep learning approaches have shown to be superior in modelling the user decision journey, which is a highly nonlinear process where touchpoints show complex interactions with each other.

Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long Short-Term Memory RNNs, and Gated Recurrent Unit RNNs were established as state-of-the-art approaches for modelling sequential data such as in the fields of natural language processing (NLP) and MTA. The sequential nature of these models precludes parallelization within training, which causes computational problems when handling long sequences due to memory constraints. 'The Transformer' (Vaswani et al., 2017), a neural network architecture that replaces recurrent layers with (self-)attention mechanisms, which was initially introduced in the field of NLP, aims to solve these computational issues. By leveraging attention mechanisms and parallel computation, significant advancements in NLP tasks have been realized, achieving state-of-the-art results. Due to its success in the field of NLP, Transformer-based architectures have been widely adopted in other research domains, such as MTA.

Over the last few years, deep learning approaches using attention mechanisms have emerged as effective and accurate solutions for tackling the complex MTA problems (Ren et al., 2018; Kumar et al., 2020; Arava, Dong, Yan, Pani et al., 2018; Li, Cheng, Chen, Chen & Wang, 2020). Even though these models greatly increase the efficiency, this comes at a cost of interpretability. In the field of MTA, interpretable models are vital when it comes to understanding which touchpoints contribute to what extent to the conversion. Therefore, a neural network architecture using a simplified attention mechanisms, which can be interpreted as conversion credit is required.

In this paper, we propose a simple, interpretable, feed-forward neural network, which we refer to as a Stacked Web of Attentional Neurons (SWAN). The architecture consists of one embedding layer, one layer of four attentional neurons, another layer of one attentional neuron and a final representation layer. The attentional neurons represent a simplified version of 'The Transformer', allowing us to trace attention weights back to individual touchpoints, facilitating interpretability. This model aims to retain the efficient properties of the aforementioned deep learning architectures, while assuring interpretability of conversion attribution at the same time. Moreover, the proposed architecture is computationally efficient, enabling training within minutes, making it highly applicable in real-world applications.

A major challenge in machine learning is the class imbalance problem, particularly in the context of MTA data. The online customer journey leads to a non-conversion outcome (majority class) significantly more frequently than compared to a conversion outcome (minority class). Training on this skewed data negatively impacts the performance of deep learning models. Models trained on highly imbalanced data tend to favor the majority class and have difficulties correctly classifying the minority class. In a dataset where 5% of the observations belong to the minority class, a high accuracy of 95% can easily be achieved by the model by solely predicting the majority class.

In the literature, oversampling is a commonly used technique to solve the class imbalance problem in deep learning. However, oversampling is a concern as models that are trained on fictitious data may fail miserably in real-world applications (Tarawneh, Hassanat, Altarawneh & Almuhaimeed, 2022). Alternatively, undersampling techniques can be used to address the class imbalance problem. A disadvantage of undersampling is the fact that large amounts of (informative) data are discarded.

Ensemble learning techniques can be leveraged to solve the class imbalance problem without losing huge amounts of information. For instance, EasyEnsemble (Liu, Wu & Zhou, 2008) creates multiple balanced datasets by randomly undersampling the majority class, and then trains multiple neural networks on the distinct balanced datasets. Before making the final classification of an input, the prediction results of all the distinct neural networks are aggregated. In this research we propose the Ensemble-SWAN, which leverages the idea of EasyEnsemble to account for the class imbalance problem present in MTA data.

Uncertainty is inseparably connected to deep learning as reliable models are critical in real-world applications. Therefore, uncertainty estimation, visualization, and quantification are hot topics in machine learning nowadays (Abdar et al., 2021). Deep learning models are often simplifications of the reality, which cause uncertainties. These uncertainties can be classified as aleatoric (due to randomness) and epistemic (due to lack of knowledge) (Rao, Kushwaha, Verma & Srividya, 2007). Epistemic uncertainty is considered to be reducible by increasing information and/or complexity, whereas aleatoric uncertainty is irreducible. In this paper, we introduce two approaches to visualize and quantify the epistemic uncertainty present in the Ensemble-SWAN. First, we visualize and quantify the epistemic uncertainty caused by the randomly undersampled datasets in the Ensemble-SWAN. We do this by evaluating the predictions of the individual constituent models of the Ensemble-SWAN, referred to as sub-NNs. Second, we use Monte Carlo Dropout (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014) to visualize and quantify the epistemic uncertainty caused by lack of model complexity present in one of the sub-NNs of the Ensemble-SWAN.

In summary, the contributions in this paper are three-fold and can be summarized as follows:

1. We propose an interpretable and computational efficient, simple feed-forward neural network using attention mechanisms for multi-touch conversion attribution problems, which we refer to as a Stacked Web of Attentional Neurons (SWAN). The model performance is evaluated on the benchmark Criteo dataset.

2. We propose the Ensemble-SWAN, an approach which accounts for class imbalance in

multi-touch conversion attribution problems by leveraging ensemble learning techniques and random undersampling.

3. We introduce two novel approaches for quantifying epistemic uncertainty in the model, pioneering advancements in the field of multi-touch conversion attribution (and click-stream data analysis in general).

All source code used in this paper can be found at `https://github.com/pimweterman/Ensemble-SWAN.git`. The remainder of this paper will be structured as follows: Section 2 discusses the academic literature related to our research. Section 3 discusses the benchmark Criteo dataset and corresponding pre-processing steps, which are necessary in order to train and evaluate the SWAN and the Ensemble-SWAN. Thereafter, Section 4 outlines the methodologies of the SWAN and the Ensemble-SWAN. Also, the two methods of the proposed uncertainty quantification (UQ) approaches are discussed. Section 5 reports the obtained results and discusses their interpretation. Finally, Section 6 summarizes the results and discusses some suggestions for future research.

2 Literature Review

Due to the major shift in advertising from offline to online, MTA modelling becomes an increasingly important topic in the literature. A considerable amount of research is recently being done on conversion prediction and conversion attribution, highlighting the significance of these fields. Multi-touch attribution is generally defined as the science of using mathematical approaches to assign conversion credit to touchpoints in a sequence of advertisements viewed by an online user (Moffett, Pilecki & McAdams, 2014). Attribution modelling enables companies to answer the critical question regarding marketing return on investment (ROI): What advertisements are driving conversions? (Kannan, Reinartz & Verhoef, 2016).

Algorithmic and Deep learning approaches will be discussed in Sections 2.1 and 2.2, respectively. Subsequently, in Section 2.3, we will discuss the class imbalance problem which is present in online marketing user journey data. Last, in Section 2.4 we will give a brief overview of UQ approaches in deep learning.

2.1 Algorithmic Approaches

In early works rule-based attribution models were developed. 'First-touch' and 'Last-touch' attribution rules are widely adopted in practice, where full conversion credit is assigned to the first and last touchpoint, respectively. However, despite the simplicity of these rule-based approaches, a disadvantage is the fact that it only recognizes the contribution of one single touchpoint in the sequence. These models do not fully capture the effects of a sequence of advertisements, as a conversion is believed to be caused by the combined effect of individual advertisements (Zhang et al., 2014). As a consequence, MTA models like Time-Decay attribution (Wooff & Anderson, 2015) and U-Shaped attribution (Buhalis & Volchek, 2021) were introduced in the literature.

In Shao and Li (2011) two data-driven approaches are proposed, a bagged logistic regression method in combination with aggregated bootstrap and a probabilistic approach based on conditional probabilities. The downside of the bagged logistic regression approach is the fact that retrieved effects are now aggregated and difficult to interpret. A solution is to derive a measure of importance for each variable for classifying a positive outcome. In Dalessandro et al. (2012) this idea is leveraged, a channel importance measure based on the Shapley value is proposed. Each touchpoint is considered as a player in a cooperative game, this logic is used to derive the Shapley value by summing the marginal contributions a touchpoint adds to all possible sequences that do not contain this touchpoint. Alternatively, Xu et al. (2016) propose a lift-based prediction model for real-time advertisement delivery, arguing that user behaviour has different additional effects on the user's conversion decision. The authors believe that the bid price (i.e., value) of an advertisement should be measured based on the performance lift among users who have and who have not been exposed to a certain advertisement.

Survival analysis-based models and techniques have proven to be a powerful tool for the analysis of conversion probability, as these techniques account for the duration and timing of user interaction with touchpoints. For example, Zhang et al. (2014) propose the Additive Hazard (AH) model, which uses an additive hazard function for conversion prediction. Likewise, Ji et al. (2016) introduce the Additional Multi-touch Attribution (AMTA) model, this model uses the hazard rate of a conversion at a specific time to model conversion attribution.

2.2 Deep Learning Approaches

More recently, deep neural network architectures have been proposed for a wide range of applications such as amongst others NLP (Goldberg, 2016), image and video recognition (Fu, Zheng & Mei, 2017), and recently MTA modelling. As a sequence of touchpoints can be seen as a sequence of words, neural NLP techniques can be leveraged to create meaningful insights in the field of MTA modelling. For example, Qu et al. (2016) propose a Product-based Neural Network (PNN) which aims to predict user response. The PNN not only explores feature interactions, but also has the ability to learn high-order latent patterns, resulting in superior performance compared to the state-of-the-art methods at that time. Similarly, Zhou et al. (2019) propose a deep learning approach to predict click-through rates. Here, a Deep Interest Evolution Network (DIEN) is proposed, where an extractor layer is used to capture temporal interest from past user behaviour.

Before the introduction of 'The Transformer (Vaswani et al., 2017), sequential data (often in the field of NLP) was often modelled by deep learning architectures like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). A downside of these models is the lack of parallelization of the input sequences during training because the output of each time step depends on the output of the previous step. By allowing the model to focus selectively on specific parts of the input sequence, Transformer-based models aim to solve the lack of parallelization. These models learn to selectively attend to relevant information from the input sequences by incorporating so-called key, value, and query computations. Because of parallelization, the model is able to capture complex and long-range dependencies in the input data.

In the literature, we have seen an enormous growth in the use of attention mechanisms in deep learning architectures since, also in the field of MTA modelling. The attention mechanisms allow the model to focus specifically on certain touchpoints in the input sequence. In Ren et al.

(2018), the Dual-attention Recurrent Neural Network (DARNN) is proposed, which learns the attribution of the conversion action over the whole sequence of touchpoints by using a Location-Base attention mechanism in a recurrent neural network. Hence, this model captures sequential user behaviour patterns and learns the attention weights using sequential modelling. Similarly, in Arava et al. (2018), a Deep Neural Net with Attention Multi-Touch Attribution model (DNAMTA) which also uses sequential modelling and the dynamic interaction effects between media channels to make conversion predictions and to evaluate media channel contributions, is proposed. Additionally, DNAMTA uses extra user information and time-decay functions to make predictions. Other approaches prove the effectiveness of deep learning architectures using attention mechanisms for prediction of conversion (Kumar et al., 2020) and prediction of click-through rate (CTR) (Li et al., 2020).

The disadvantage of these relatively complex neural network architectures is the lack of interpretability. In most Transformer-based attention mechanisms it is not possible to trace back attention weights to individual tokens in the input sequence. In Raffel and Ellis (2015), a simplified form of attention, where the attention weights are directly learned from the encoder hidden-states, which allows for variable length input sequences and can handle input sequences longer than the ones present in the training set, is introduced. This model produces a single context vector from an entire input sequence. Hence, enabling us to trace back attention weights to individual tokens in the input sequence. In the field of MTA, this simplification is justified as there is no need to model complex language relations

In this paper, we introduce a relatively simple feed-forward neural network approach using simplified attention mechanisms (Raffel & Ellis, 2015) which can match state-of-the-art accuracy results while maintaining interpretability.

2.3 Class Imbalance

Not only in the context of MTA, but in the entire field of deep learning, the class imbalance problem (Japkowicz & Stephen, 2002) is a major issue that needs solving. This facilitated the birth of widely-used oversampling techniques like Synthetic Minority Over-sampling Technique (SMOTE) (Chawla, Bowyer, Hall & Kegelmeyer, 2002) and Adaptive Synthetic Sampling (ADASYN) (He, Bai, Garcia & Li, 2008). These techniques aim to solve the class imbalance problem by generating synthetic observations of the minority class until the dataset is balanced. Several variants of SMOTE, such as Borderline-SMOTE (Han, Wang & Mao, 2005) and Safe-level-SMOTE (Bunkhumpornpat, Sinapiromsaran & Lursinsap, 2009) are also widely-used in practice.

Unfortunately, oversampling does not come without risks. In Tarawneh et al. (2022), the authors critically review over 70 oversampling techniques and conclude that models trained on fictitious data may fail miserably when used in real-world applications. They argue that the fundamental challenge with oversampling approaches is the fact that synthetically generated data points may actually not belong to the minority class in the real-life population. Surprisingly, the authors even demonstrate that all reviewed methods generate minority data points that are highly likely to belong to the majority class in real-life. Hence, the authors manifest to stop oversampling for class imbalance learning.

Alternatively, to reduce the risk of failures in real-world applications, undersampling techniques can be used. For instance, Tomek's link undersampling (Tomek, 1976) finds pairs of observations that are nearest neighbours, one from both the minority and majority class, and then discards the majority class instance. Cluster centroids undersampling (Lemaître, Nogueira & Aridas, 2017) uses clustering techniques to remove observations from the majority class while preserving the distribution and structure of the dataset. A downside of these techniques is the fact that data instances and thus valuable information are discarded.

Ensemble learning techniques provide a solution for the loss of information caused by undersampling. As mentioned in the introduction, the idea of EasyEnsemble (Liu et al., 2008) can be leveraged to obtain class balance by undersampling while avoiding throwing away too much valuable information. In this paper, we introduce Ensemble-SWAN, a method that trains multiple neural networks on separate randomly undersampled datasets, and subsequently aggregates predictions before classification.

2.4 Uncertainty Quantification (UQ)

With the growing adoption of deep learning solutions in real-world applications, the correct quantification of uncertainties in processes and predictions is crucial (Jiang, Kim, Guan & Gupta, 2018). This need facilitated the birth of various UQ techniques for deep learning solutions in the field of NLP (e.g., machine translation), medical image analysis (e.g., medical image classification), and computer vision (e.g., self-driving cars). A comprehensive overview of these techniques can be found in Abdar et al. (2021). The fact that between 2010 and 2021 more than 2500 papers addressing UQ in the field of AI were published, underscores the importance of this topic.

As mentioned in the introduction, there are two types of uncertainty: aleatoric and epistemic (Hüllermeier & Waegeman, 2021). Aleatoric uncertainty (also known as data uncertainty) is irreducible and is often attributed to measurement errors, noise, or natural stochasticity. On the other hand, epistemic uncertainty (also known as model uncertainty) is caused by the limited amount of knowledge and arises from the model's limited understanding of the underlying system. Epistemic uncertainty can be reduced with additional information and improved model fit.

Bayesian approaches, like Bayesian NNs (BNNs) (Izmailov, Vikram, Hoffman & Wilson, 2021), provide a mathematical framework for reasoning under uncertainty and can help quantifying uncertainty in model predictions. Unfortunately, Bayesian approaches often come with prohibitive computational cost. In Gal and Ghahramani (2016), the authors show that the use of Monte Carlo Dropout in neural networks can be interpreted as a Bayesian approximation of the Gaussian Process (GP) probabilistic model (Rasmussen, Williams et al., 2006). Normally, dropout is a regularization technique that is used in more complex neural networks to prevent overfitting. However, Monte Carlo Dropout has also proven to be an effective and computationally efficient tool for estimating and visualizing epistemic uncertainty in neural networks.

In addition to Bayesian approaches, ensemble learning techniques have been widely-used to quantify uncertainty in deep neural networks. In the literature it is shown, that training multiple neural networks independently and aggregating the predictions can enhance performance and quantify uncertainty (Fort, Hu & Lakshminarayanan, 2019). A disadvantage of ensemble learning is the fact that multiple computationally inefficient models need to be trained. To address this issue, efficient ensembles of deep neural networks are developed for a broad spectrum of applications (Egele et al., 2022; Wen, Tran & Ba, 2020; Vallabhajosyula, Sistla & Kolli, 2022). Due to the computational simplicity of our proposed approach, we can leverage an approach based on classical, computational 'infeasible' ensemble techniques for UQ, like Bagging (Breiman, 1996; Dietterich, 2000) and Bayesian averaging (Raftery, Madigan & Hoeting, 1997).

3 Data

In this Section, we will discuss the benchmark Criteo dataset used in this research. First, we give a brief overview of the data in Section 3.1. Next, we will discuss the necessary pre-processing steps in Section 3.2. Last, we will briefly touch on the class imbalance problem present in the Criteo dataset in Section 3.3.

3.1 Data Overview

In this research we will use a dataset from **Criteo**, a leading company in online marketing and advertising research. The Criteo dataset is being utilized in this study as it is considered the benchmark dataset in the field of MTA, thereby enabling us to compare our outcomes to state-of-the-art models. The core business of Criteo is selling advertisements via display. Criteo published a dataset for attribution modelling in real-time auction based advertising (Diemert, Meynet, Galland & Lefortier, 2017). In this dataset, useful information on advertisement exposure and clicking behaviour is captured, which can be used for online user behaviour analysis. The raw dataset consists of over 16 million touchpoints over 675 campaigns, captured from Criteo live traffic in a period of 30 days. This results in 6.1 million touchpoint sequences, of which approximately 550,000 led to a conversion. Each touchpoint in the dataset is characterized by 9 categorical variables as presented in Table 1, which consist of 59,098 unique (sub)categories, of which the context is masked to remain confidentiality. These category IDs contextualize the advertisement by including characteristics of the advertisement, such as the device where the advertisement is shown on.

Table 1: Categories and amount of sub-categories characterizing touchpoints in the Criteo dataset.

Categorical variable	Unique values
Campaign ID	675
Category 1	9
Category 2	70
Category 3	1,829
Category 4	21
Category 5	51
Category 6	30
Category 7	57,196
Category 8	11
Category 9	30

For every shown advertisement the following information is captured: relative timestamp, if the advertisement was clicked, if the advertisement led to a conversion, unique user ID, unique campaign ID, timestamp of conversion, number of clicks, time since last click given the advertisement, and category IDs. Additionally, the dataset contains information on the price paid for displaying the advertisement and the monetary order size if converted. This monetary data will not be used in this research, thus can be disregarded. Please find all variables and more detailed descriptions in Table 2.

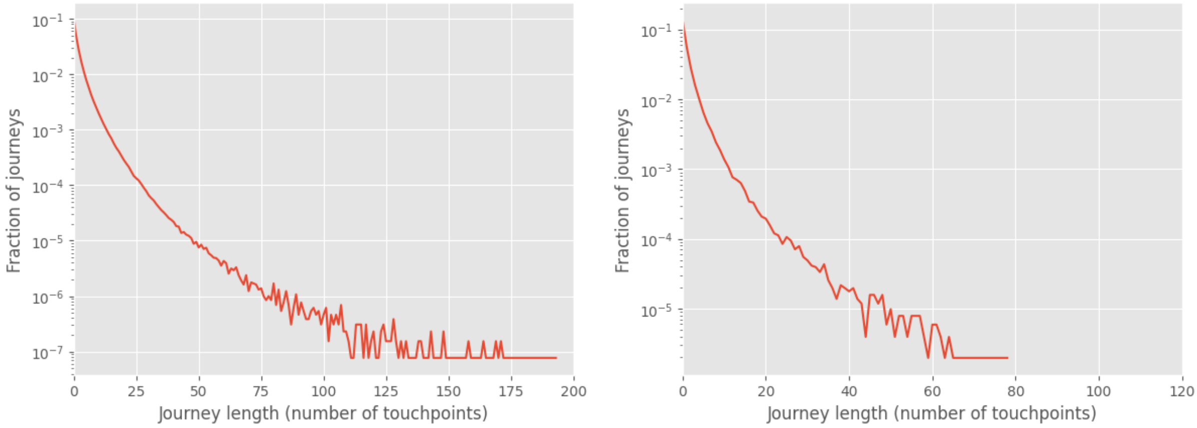
Table 2: Variables and corresponding descriptions captured in Criteo dataset.

Variable	Description
timestamp	Timestamp of the impression (starting from 0 for the first impression)
uid	Unique user identifier
campaign	Unique campaign identifier
conversion	1 if there was a conversion in the 30 days after the impression, 0 otherwise
conversion_timestamp	Timestamp of the conversion, -1 if no conversion was observed
conversion_id	Unique conversion identifier, -1 if no conversion was observed
click	1 if the impression was clicked, 0 otherwise
cost	Price paid by Criteo for the display
cpo	Cost-per-order in case of attributed conversion
cat[1-9]	Contextual features associated to the display. Each column is a categorical variable representing contextual features such as browser, device, format etc. They are mapped to a fixed dimension space using hashing trick.

3.2 Data Cleaning and Pre-Processing

In order to input the data in the SWAN and Ensemble-SWAN, some data cleaning and data pre-processing steps are required. Following the data pre-processing steps of Ren et al. (2018), the raw data can be transformed into sequences of touchpoint per user. Each user can be associated with a single conversion ID or multiple conversion IDs, hence we split the sequences based on conversion time in such a way that we have at most one conversion per sequence. Additionally, sequences with less than three touchpoint are removed from the dataset as no useful information on browsing behaviour can be derived from these sequences. From Figure 2 we see that the shorter the user journey, the higher the fraction of conversions is. Taking into account computational feasibility of the pre-processing steps, we decide that a maximum journey length of 20 touchpoints suffices in this research. After performing the above pre-processing steps, we obtain a clean dataset with a conversion ratio of approximately 4.7%.

We split the clean dataset in a training set and a test set, containing 80% and 20% of the sequences, respectively. The training set consists of 287,145 sequences and the test set of 71,787 sequences, both maintaining a conversion ratio of approximately 4.7%. To efficiently input the data in the SWAN and Ensemble-SWAN we create three-dimensional batches of size (*batch size*, *maximum sequence length*, *touchpoint characteristics*) = (1024, 20, 13). When sequences are shorter than the maximum length of 20 touchpoints, we set all the remaining values equal



(a) Journey length all journeys

(b) Journey length converted journeys

Figure 2: Fraction of total and converted sequences with respect to sequence length.

to zero.

3.3 Class Imbalance

After pre-processing, the remaining 287,145 sequences in the training set have a conversion ratio of approximately 4.7%. Hence, the dataset used in this research is highly imbalanced.

State-of-the-art neural models for MTA problems strictly employ oversampling of the minority class techniques to reach class balance. To compare the SWAN with other models such as the ARNN proposed in Ren et al. (2018), we will also train one instance of the neural network using a balanced dataset obtained by oversampling the minority class (i.e., conversion) using ADASYN (He et al., 2008). This results in a training set of 530,145 sequences with a conversion ratio of 48%. The test set will not be balanced.

As discussed in Section 1, we believe that machine learning architectures trained on oversampled data can fail miserably in real-world applications. To overcome this we introduce the Ensemble-SWAN, where distinct neural networks are trained on randomly undersampled datasets before aggregating the results. All of these randomly undersampled datasets consists of 27,348 sequences with a conversion ratio of 50%. Again, the test set will not be balanced.

4 Methodology

First, in Section 4.1, we will describe the proposed feed-forward neural network with simplified attention mechanism called Stacked Web of Attentional Neurons (SWAN) in great detail. Second, in Section 4.2, we will describe the Ensemble-SWAN, our approach that leverages ensemble learning techniques to compensate for the information loss caused by random undersampling. Third, to address the need for acknowledging uncertainty in real-world applications, we explore and discuss two approaches to visualize and quantify uncertainty present in the Ensemble-SWAN in Section 4.3.

4.1 SWAN

This Section provides a comprehensive discussion of the various components comprising the SWAN. We will start by discussing the attention mechanisms in Section 4.1.1. As this attention mechanism is order-agnostic, we will add positional encodings to the input embeddings, which are described in Section 4.1.2. Next, to form the SWAN, we stack attentional neurons, this is described in Section 4.1.3. Then, in Section 4.1.4, we describe how the final representation is converted to a conversion probability. For completeness, we will add some detail regarding model settings and specification in Section 4.1.5. Last, in Section 4.1.6, we will discuss various performance evaluation metrics which are used to assess the performance of the SWAN and Ensemble-SWAN.

4.1.1 Attention Mechanism

In the past, Recurrent Neural Networks (RNNs) were often used for modelling sequential data as they are able to model temporal dependencies in sequences such as sentences. However, during training of RNNs with backpropagation challenges such as the vanishing and exploding gradient problem arise (Pascanu, Mikolov & Bengio, 2013). Resulting in the fact that, RNNs are almost exclusively used for tasks where the sequential dependencies span across a large number of time steps. Moreover, as evaluation of RNNs is sequential and cannot be parallelized, the training process can become computationally inefficient for long sequences. Attention mechanisms were introduced to solve the problems stated above.

In our proposed neural network architecture we make use of a simple attention mechanism as proposed in Raffel and Ellis (2015). This architecture produces a single context vector $\boldsymbol{c}$ from an entire user journey input sequence consisting of multiple touchpoints. Each touchpoint is embedded in an input vector z_i. The proposed attention mechanisms weighs the importance of every touchpoint in the user journey and computes importance values r_i which are transformed accordingly to attention weights v_i using the softmax function. Finally, the context vector $\boldsymbol{c}$ is computed as the weighted sum of the attention weights v_i and the original input vectors z_i.

The simple feed-forward attention mechanisms can be described by the following three equations.

$$r_i = \alpha(z_i) = {u_\alpha}^T tanh(W_\alpha z_i + b_\alpha) \tag{1}$$

$$v_i = softmax(\mathbf{r}) = \frac{exp(r_i)}{\sum_j exp(r_j)} \tag{2}$$

$$\mathbf{c} = \sum_i v_i z_i \tag{3}$$

The embedding vector z_i has dimension d In Eq. 1 the learnable function $\alpha(z_i)$, which merely depends on input vector z_i, computes the importance values. In this function, matrix W_α of dimension d_h x d, and vectors b_α and u_α of dimension d_h x 1, are learnable parameters. The attention weights computed in Eq. 2 by the softmax activation function, form a probability distribution that tells the model to what extent a touchpoint contributes to the conversion or non-conversion, when predicting the outcome of a sequence. Last, in Eq. 3 the output of the attention mechanism, the fixed-length context vector $\boldsymbol{c}$, is computed by the weighted average of the attention weights v_i and input vectors z_i.

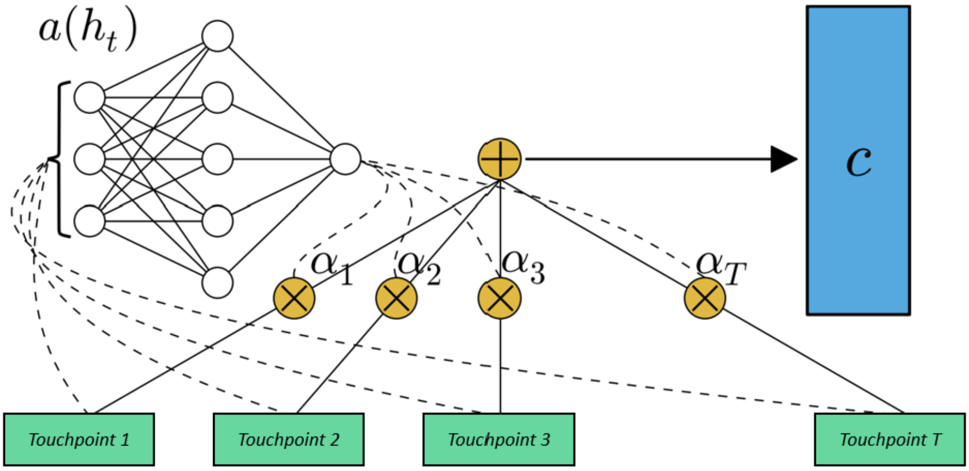


Figure 3: The architecture of an attentional neuron as proposed in Raffel and Ellis (2015).

The architecture in Figure 3 illustrates the fundamental idea of the attention mechanism described above. When a certain touchpoint in a user journey sequence has a high attention weight, we can say that this touchpoint significantly contributed to the conversion prediction.

4.1.2 Positional Encodings

Attention mechanisms, like ‘The Transformer (Vaswani et al., 2017) and the attentional neuron used in the proposed architecture, are order-agnostic (Yun, Bhojanapalli, Rawat, Reddi & Kumar, 2019). Unlike in Recurrent Neural Networks (RNNs), the attention mechanism process the input sequence in parallel, causing the model to lack the inherent notion of order. To explicitly encode positional and order information in the model, positional encodings are employed.

Absolute positional encodings using sinusoidal functions are the most standard and effective way of incorporating positional information in the attention mechanism and was first introduced in Vaswani et al. (2017). The idea is to construct an encoding vector of dimension d_h by using sine and cosine functions of different frequencies. The positional encoding vector for the i-th touchpoint in the sequence is defined as follows:

$$p_i = \begin{cases} \sin(pos/10000^{(2i/d_h)}) & \text{if} i = 2i \\ \cos(pos/10000^{(2i/d_h)}) & \text{if} i = 2i+1 \end{cases} \tag{4}$$

where pos is the position of the touchpoint in the sequence i = 1, ... $d_h/2$. The positional encoding vector consists of y-coordinates from sinusoidal functions of different wavelengths as defined above, evaluated at a x-coordinate that depends on pos. These positional encodings are added to the input vectors z_i. The advantage of the sinusoidal encodings is the fact that the functions have a smooth and continuous path, which enables the model to extrapolate to sequence lengths longer than encountered in training.

4.1.3 Stacked Web of Attentional Neurons (SWAN)

When using a single attention mechanism as described in Section 4.1.1, the importance score given to a touchpoint in the user journey always stays the same regardless of other touchpoints in the user journey, assuming that the touchpoint occurs at the same position in the user journey. To capture complex interactions between touchpoints in the user sequence we use a stacked web of attention mechanisms. Different neural networks $\alpha_l(.)$ are used in the architecture to compute multiple context vectors c_l. Ultimately, the final representation $\boldsymbol{c}$ is computed by a single attentional neuron layer, which is computed as the weighted sum of the context vectors outputted by the different neural nets $\alpha_l(.)$:

$$\boldsymbol{c} = \sum_l v_l \sum_i v_{li} z_i' \tag{5}$$

In Eq. 5, v_{li} is the attention given to the i-th touchpoint in the sequence by the l-th context vector. Whereas, v_l is the attention given to the l-th context vector by the final representation. The total attention given to the i-th touchpoint in the user sequence can then be computed as follows:

$$v_i = \sum_l v_l v_{li} \tag{6}$$

The total attention for each touchpoint v_i is automatically dependent on the entire input sequence as v_l are outputs from neural networks $\alpha_l(.)$, which take the entire sequence as input.

Stacking multiple attention mechanisms as described above, results in a model architecture we call a Stacked Web of Attentional Neurons (SWAN). The architecture is schematically presented in Figure 4. In our proposed model we use a layer of four attentional neurons and a layer of a single attentional neuron between the embedding and final representation layer.

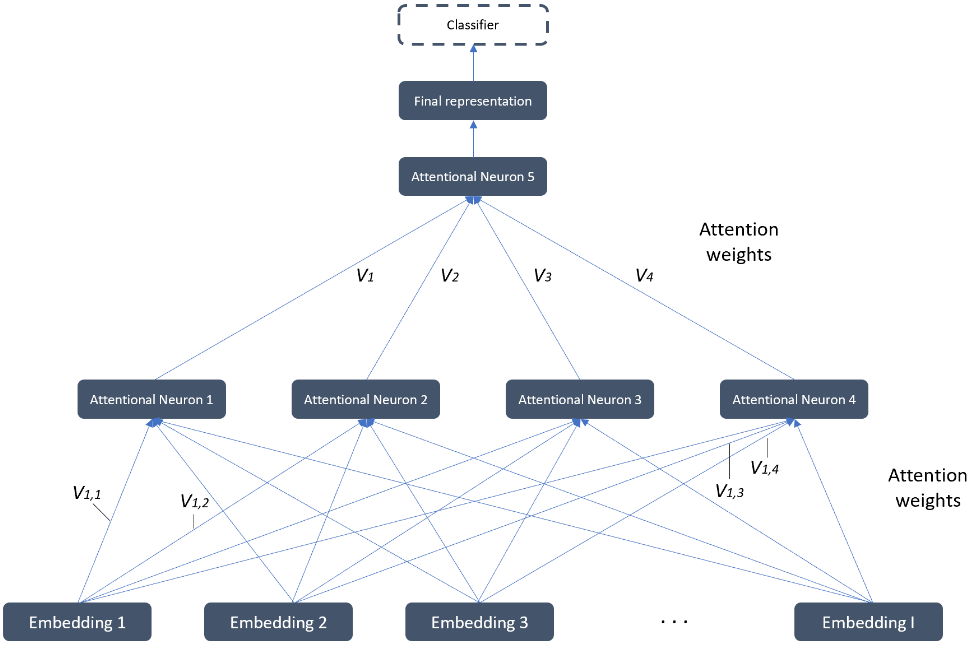


Figure 4: The architecture of the Stacked Web of Attentional Neurons (SWAN).

4.1.4 Conversion Prediction

The final representation $\boldsymbol{c}$, as defined in Eq. 5, is used to make a final prediction of the probability of conversion for a particular sequence. We transform $\boldsymbol{c}$ to a conversion probability as follows:

$$P(Y = 1 \mid \boldsymbol{c}) = sigmoid(\phi(W_c\mathbf{c}) + b_c) \tag{7}$$

Where sigmoid is a logistic function, sigmoid$(x) = 1/(1 + e^{-x})$. The Rectified Linear Unit (ReLU) activation function is defined as follows $\phi = max(0, x)$. We classify a sequence as a conversion when $P(Y = 1 \mid \mathbf{c}) > 0.5$

The sigmoid function transforms the final representation $\boldsymbol{c}$ to a number between 0 and 1, allowing the resulting value to be interpreted as a probability. Moreover, due to its characteristics, the ReLU activation ensures that the touchpoints in the sequence can solely make a positive contribution on the conversion probability.

4.1.5 Model Details

As we normalize the data instead of using one-hot encoding schemes, the embedding vectors of the input data have dimension $d = 12$. All feed-forward neural networks with simple attention mechanisms consists of one layer of hidden size $2 \times d$. We use four attentional neurons as described in Section 4.1.1 between the input and final representation layer. Another single attentional neuron layer is used before computing the final representation. To model complex and non-linear relations and dependencies, we use the hyperbolic tangent activation function to activate the neurons in the attention mechanisms. Optimization during training is carried out using the binary cross-entropy (BCE) loss criterion and the Adam optimizer, which adaptively adjust learning rates for the model parameters to reach smooth convergence. The learning rates of the Adam optimizer are initialized as follows: $\alpha = 1 \times 10^{-2}$, $\beta_1 = 0.90$, $\beta_2 = 0.98$ and $\epsilon = 1 \times 10^{-9}$. We allow the model to learn itself the importance of the positional encodings by introducing learnable parameter γ, for which we choose a higher learning rate of 0.01. The neural network is implemented using Python's machine learning library PyTorch (Paszke et al., 2019) and trained on a Asus Vivobook with a 12th Gen Intel(R) Core(TM) i7-12700H processor and 16GB memory.

4.1.6 Performance Evaluation Metrics

As this research is focused on the implications of real-world application of the SWAN and Ensemble-SWAN, we will evaluate the performance using several performance measures. As SWAN and Ensemble-SWAN can be utilized by diverse stakeholders (marketers, advertisement auctioneers, etc.), who all have different objectives. Hence, it is of great importance to evaluate the SWAN on different aspects. In this research we will use 5 performance measures: Accuracy, Precision, Recall, F-Measure and Area Under the Receiver Operating Characteristics Curve (AUC-ROC).

We will use the following abbreviations to define the performance measures: TP = True Positive, TN = True Negative, FP = False Positive, & FN = False Negative.

The most widely-used performance measure Accuracy is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Accuracy can be a misleading performance measure in datasets where the class imbalance problem is present. For example, in our dataset we have a conversion ratio of approximately 4.7%, when the model only predicts non-conversion a high accuracy of roughly 95% will be achieved. Nonetheless, accuracy remains a valuable performance measure to report due to its generalizability.

Precision is defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Precision tells us how many of the predicted converting sequences are actually a converting sequence. Precision is a particularly good measure to use when the cost of a False Positive are high.

Recall is defined as follows:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

Recall tells us how many of the actual conversions in the dataset are classified as a conversion by the model. Recall is often used in cases where there are high cost associated with a False Negative.

The F1-Score is a combination of Precision and Recall and is defined as follows:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

The F1-Score is used when we seek a performance measure that balances the importance of Precision and Recall. The F1-Score is particularly suitable in situations where we have a highly imbalanced dataset and where accurately identifying both positive and negative instances is equally important. In the subsequent uncertainty analysis we will use the F1-Score as leading metric when performance results will be reported. Please note that in this paper we use the weighted variants of Precision, Recall, and F1-Score as these metrics assign higher importance to certain classes to create a more balanced evaluation metric in the case of class imbalance.

Another performance measure that is particularly popular when evaluating the performance of a binary classification task is the AUC-ROC (Davis & Goadrich, 2006). The AUC-ROC tells us to what extent the model is capable of distinguishing between a converting and a non-converting sequence. The higher the AUC-ROC, the more capable the model is. The AUC-ROC is especially useful in settings where the class imbalance problem is present, as the AUC-ROC provides a more comprehensive evaluation of discriminatory power rather than simply assessing the overall correctness of predictions, as for example accuracy does.

4.2 Ensemble-SWAN

In order to compare the SWAN to a state-of-the-art MTA deep learning solution, we first test the model on a balanced dataset generated by oversampling using ADASYN. However, as mentioned in Section 1, we believe that training a model on fictitious, oversampled data can fail miserably in real-world applications. To solve this problem, we propose an ensemble deep learning approach using random undersampling, called Ensemble-SWAN. This model avoids generating fictitious data through oversampling, instead creating balanced datasets through undersampling. To overcome the drawback of discarding valuable data, we create N randomly undersampled datasets. Then, we train N neural networks, referred to as sub-NNs, using these datasets, and aggregate the conversion probabilities before classifying instances as a conversion or non-conversion. This approach will robustify the model predictions, as the model will be less sensitive to inconsistencies in the relatively small undersampled datasets. The ensemble learning approach is described in Algorithm 1.

Algorithm 1 Ensemble-SWAN

```
    Input: $N$ = number of balanced datasets, epochs
    Output: predictions $y$, evaluation metrics
for each $N_i$ in $N$ do
    Create 50/50 balanced dataset by random undersampling
    Divide dataset in batches of 1024, maintaining 50/50 class balance
    for each epoch in epochs do
        for each batch in $N_i$ do
            Train SWAN on balanced batch and obtain conversion probabilities $p_i$
Average the conversion probabilities over $N$ neural networks
Convert averaged conversion probability to conversion classification {0, 1}
Calculate performance evaluation metrics
Return predictions $y$, evaluation metrics
```

4.3 Epistemic Uncertainty Quantification

To be able to use deep learning solutions for real-world applications, it is vital to quantify the uncertainty present in the model. In this Section, we assess the epistemic uncertainty in the Ensemble-SWAN. We will quantify and visualize two different types of epistemic uncertainty. First, in Section 4.3.1, we will analyze the uncertainty caused by training multiple neural networks on relatively small datasets in the Ensemble-SWAN. Second, in Section 4.3.2, we will analyze the uncertainty in one of the sub-NNs of the Ensemble-SWAN by using UQ technique Monte Carlo Dropout.

4.3.1 Uncertainty: Ensemble Approach

To prevent too much information loss by random undersampling when balancing the datasets, the Ensemble-SWAN is proposed in Section 4.2. As explained previously, the Ensemble-SWAN aggregates the conversion probabilities of N neural networks before classifying a sequence as conversion or non-conversion. To get some more insights in the inner workings of the Ensemble-SWAN we will visualize the probability distributions of the conversion probabilities of the sub-NNs of the Ensemble-SWAN. This allows us to assess to what extent the conversion probabilities are subject to differences in the randomly undersampled datasets on which the sub-NNs are trained. Moreover, we assess the model performance of the sub-NNs by calculating the F1-

Score for every sub-NN, before aggregating the results in order to show the variability of model performance caused by the random undersampling when no aggregation is applied.

4.3.2 Uncertainty: Monte Carlo Dropout

Another popular technique to quantify epistemic uncertainty in neural networks is Monte Carlo Dropout (Gal & Ghahramani, 2016). Normally, dropout regularization (Srivastava et al., 2014) is a technique used to prevent overfitting when training more complex neural network architectures. By randomly disabling a fraction of the nodes during training, the network is forced to learn redundant representations and to reduce its reliance on specific nodes in the network, which helps improve generalization capability of the model.

In Gal and Ghahramani (2016), the idea of dropout regularization is extended to the testing phase in order to quantify epistemic uncertainty present in the neural network. Instead of using a single forward pass through the network when calculating conversion probabilities, Monte Carlo Dropout performs n_{MCD} forward passes with dropout enabled. Concretely, this means that during every forward pass of the data during the testing phase, a different set of nodes is randomly dropped out, which results in slightly different predictions. The idea is to generate a distribution of possible outcomes rather than just predicting a single value. Plotting this probability distribution for certain sequences of the input data, allows us to visually assess the uncertainty regarding the conversion probability predictions. Also, aggregating the predictions results from different forward passes during testing robustifies the model's predictions. In this research, we will merely use Monte Carlo Dropout to visualize the probability distribution of predictions and to calculate the variance of the predictions. The Monte Carlo Dropout UQ technique is described in Algorithm 2.

Algorithm 2 Monte Carlo Dropout

```
    Input: Pre-trained model $M_{trained}$, Number of samples $n_{MCD}$, Test data $X_{test}$
    Output Model predictions $y_{pred}$
for each i in range($n_{MCD}$) do
    Dropout regularization ($p = 0.25$)
    Run test set through model $y_{pred} = M_{trained}(X_{test})$
Plot prob. distribution $y_{pred}$
Return $y_{pred}$
```

We perform the Monte Carlo Dropout uncertainty analysis for $n_{MCD} = 1000$ forward passes and use a dropout percentage of 25%.

5 Results

The results presented and discussed in this Section are three-fold. First, in sake of comparison, we evaluate the performance of the proposed SWAN for a balanced by oversampling Criteo dataset in Section 5.1.1. Additionally, we will provide insights of the inner workings of the SWAN and demonstrate the model's ability to assign conversion credit to individual touchpoints in the input sequence. Second, in Section 5.1.2 we evaluate the performance of the Ensemble-SWAN, an ensemble learning approach which trains multiple neural networks on distinct, randomly undersampled datasets. Third, in Section 5.2 we present results of two approaches used to visualize and quantify epistemic uncertainty in the Ensemble-SWAN: Ensemble Approach and Monte Carlo Dropout.

5.1 SWAN

First, to compare the SWAN with another state-of-the-art MTA neural network solution as proposed in Ren et al. (2018), we first evaluate the SWAN on a Criteo dataset which is balanced by oversampling technique ADASYN in Section 5.1.1.

Next, as we believe oversampling can fail miserably in real-world applications, we evaluate the performance of our undersampling ensemble learning approach in Section 5.1.2. Here, we train multiple neural networks on balanced Criteo datasets, which are obtained by random undersampling, and aggregate the predictions before classification.

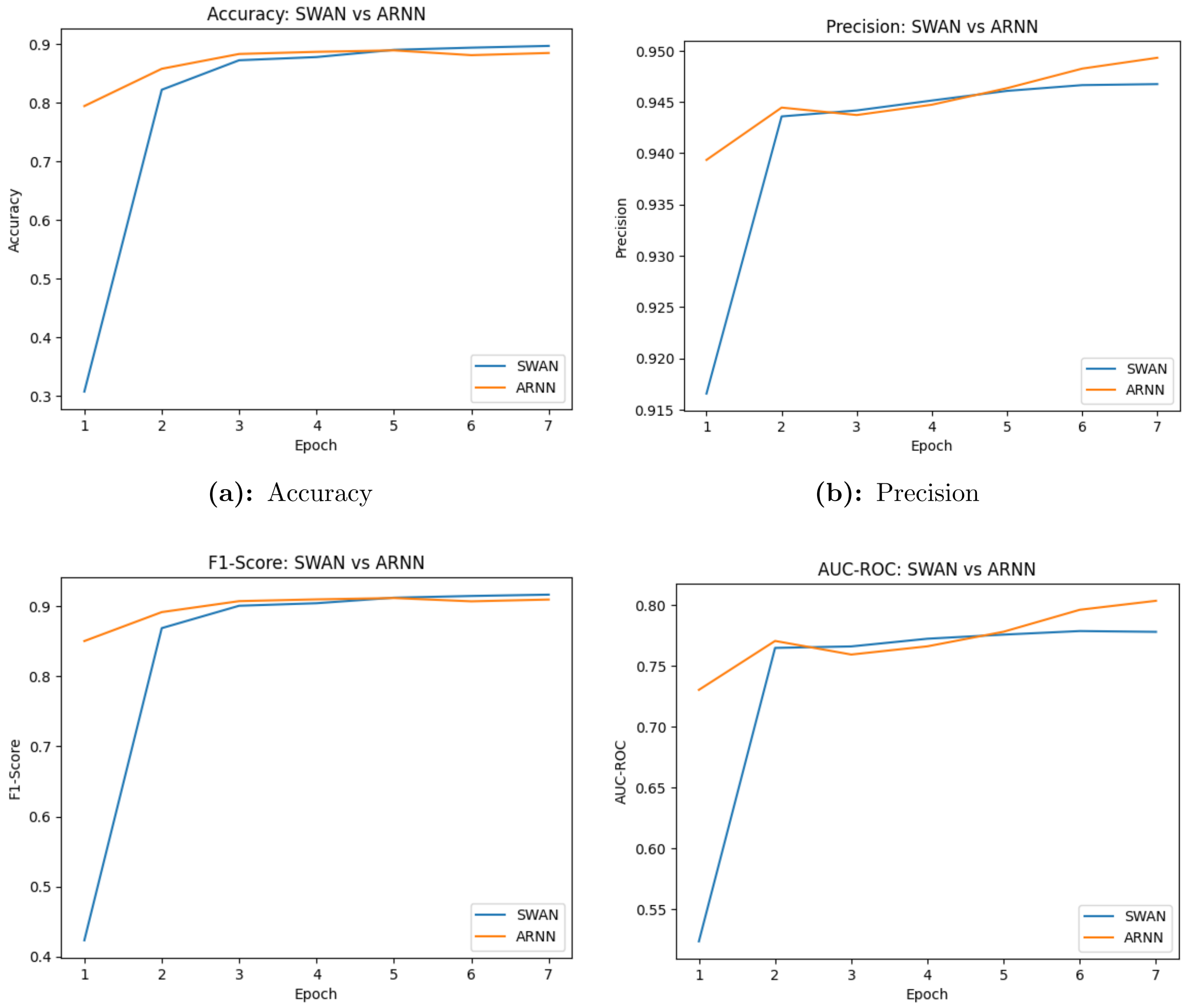
5.1.1 SWAN (Oversampling)

Most state-of-the MTA solutions account for the class imbalance problem by oversampling the dataset using various oversampling techniques. Therefore, we balance the Criteo dataset using oversampling technique ADASYN to compare the SWAN to a RNN-based attribution model as proposed in Ren et al. (2018). Approximately 4.7% of the sequences in the raw Criteo dataset lead to a conversion, after oversampling this ratio increases to roughly 48%. To mitigate memory issues, we split the data in batches of 1024 while maintaining a constant conversion ratio of roughly 48% in all batches. Consequently, both models are trained for 7 epochs using a learning rate of $\alpha = 1 \times 10^{-2}$.

Given the variability in user goals and objectives of the SWAN, we assess the performance using different performance evaluation metrics as discussed in Section 4.1.6. The results are summarized in Figure 5. In Figure 5(a) we see that both models achieve reasonably high out-of-sample accuracies of approximately 89.7%. Moreover, we see that the SWAN slightly outperforms the ARNN after 7 epochs. The weighted precision of both models is around 94.5% and depicted in Figure 5(b). Here, we see that the ARNN slightly outperforms the SWAN, meaning that the ARNN is slightly more accurate in correctly classifying converting sequences as a conversion. Please note that we are not reporting the results of the weighted recall as it is exactly equal to the accuracy in our case. This indicates that all converting sequences are correctly classified. In Figure 5(c) we observe that both models demonstrate comparable performance regarding the weighted F1-Score, which converges at approximately 91.7%. Last, Figure 5(d) reveals that the ARNN exhibits slightly superior performance after 7 epochs compared to the SWAN model regarding the AUC-ROC. The SWAN achieves a AUC-ROC of 77.8%, while the ARNN achieves a slightly higher value of roughly 80%.

Given all of the above, we can conclude that the SWAN shows similar performance as the ARNN, despite having a much simpler architecture. Moreover, due to its simpler architecture, the SWAN can be trained within minutes and is therefore computationally more efficient than the ARNN.

Another advantage of the SWAN is its ability to trace back attention weights to individual tokens in the input sequence. Hence, allowing the user to assign conversion credit to individual



(a): Accuracy

(b): Precision

(c): F1-Score

(d): Area-Under-Curve (AUC)

Figure 5: SWAN vs ARNN: Accuracy, Precision, F1-Score, and AUC-ROC.

touchpoints in the online user sequence. In Figure 6, a heatmap representation of the attentional distribution is provided, giving insights into the inner mechanisms of the SWAN. The heatmap depicts an exemplary converting user sequence with a (maximum) length of 20 touchpoints.

In this illustration it is clear that each attentional neuron captures a different context of the input sequence. The first and the second attentional neuron mostly attend to touchpoints somewhere in the middle of the input sequence. The third attentional neuron reveals a significant focus on the last touchpoint within the input sequence, whereas the fourth attentional neuron mainly focuses on the first few touchpoints.

As described in Section 4.1.3, the resulting context vectors from the four attentional neurons are again weighted through a fifth attentional neuron. These attention weights v_l are depicted on the right side of Figure 6. Here, we see that more than half of the attention is directed to the second attentional neuron ($v_2 = 0.53$). The third and the fourth attentional neuron jointly receive almost half of the remaining attention ($v_3 = 0.25$ and $v_4 = 0.18$). The context vector resulting from the fourth attentional neuron gets almost no attention ($v_4 = 0.04$), which makes sense as a single touchpoint in the middle of the input sequence often does not greatly contribute to the conversion decision. As described in Eq. 11, the SWAN architecture allows to trace back attention distribution to an individual touchpoint in the input sequence. The attention distribution of the final representation is illustrated in the last heatmap of Figure 6.

In this case, for predicting the conversion, the model attends somewhat to the begin, the middle and the last touchpoint in the user sequence. However, please note that the inner mechanisms of the SWAN differ depending on the characteristics of the input sequence. For other input sequences, the model will focus on the beginning, the middle, or the end of the input sequence, or a combination of those three. This flexibility, allows the SWAN to account for differences in user behaviour.

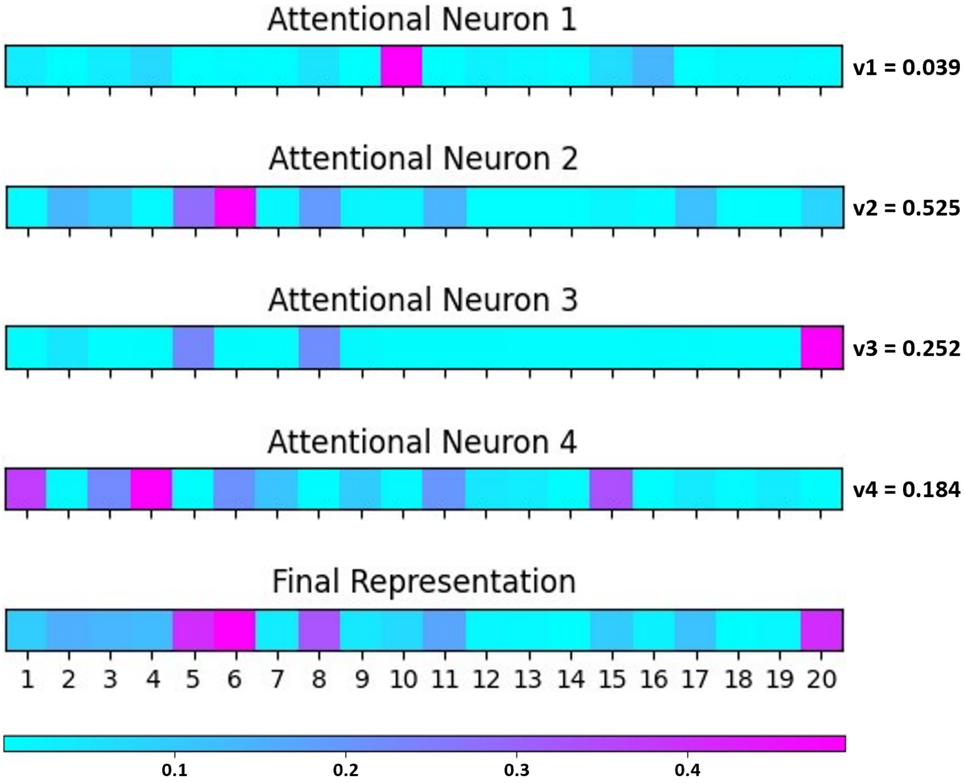
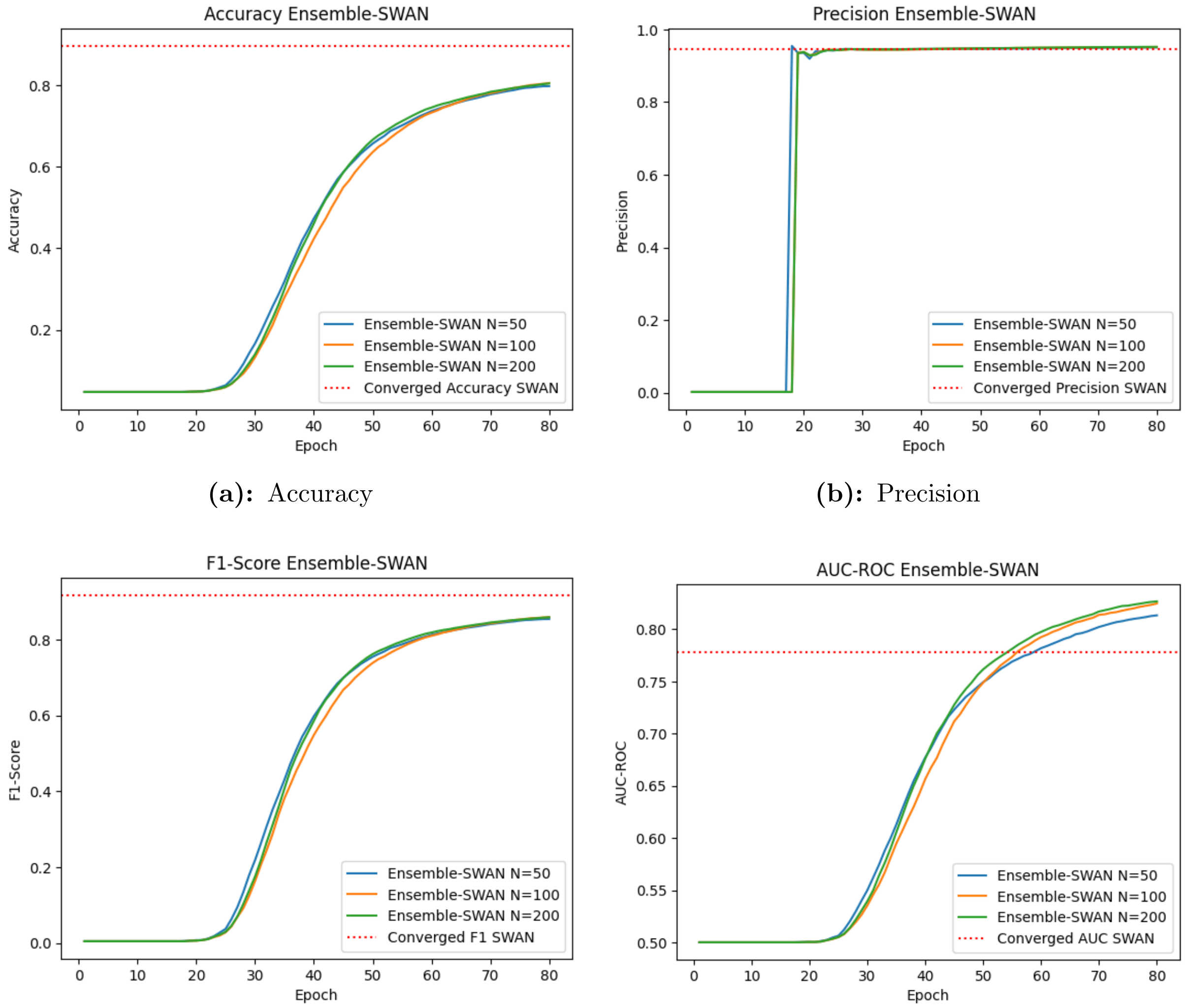


Figure 6: Inner workings SWAN: Attention weights of touchpoints for an exemplary converting sequence of length 20.

5.1.2 Ensemble-SWAN (Undersampling)

In this research, we propose a new ensemble learning solution for MTA problems as described in Section 4.2. The performance of the Ensemble-SWAN will again be assessed using the performance evaluation metrics as described in Section 4.1.6. The N undersampled, balanced datasets all consist of 27,348 sequences of which 13,674 lead to a conversion, and are therefore significantly smaller than in the previous analysis. Hence, training the Ensemble-SWAN requires a higher amount of 80 epochs and a slightly lower learning rate of $\alpha = 5 \times 10^{-3}$. Again, for computational reasons, we split the data in batches of 1024 sequences, while maintaining a conversion ratio of 50%.

The out-of-sample performance results for different levels of N are summarized in Figure 7. Please note that we only show the convergence level of the SWAN as this model is trained



(a): Accuracy

(b): Precision

(c): F1-Score

(d): Area-Under-Curve (AUC)

Figure 7: Ensemble-SWAN vs SWAN: Accuracy, Precision, F1-Score, and AUC-ROC. N is the number of sub-NNs in the Ensemble-SWAN.

on 7 epochs instead of 80, therefore difficult to jointly visualize in one figure. In this comparative analysis, three Ensemble-SWAN models with varying sizes (N=50, N=100, N=200) are evaluated, with particular emphasis on the N=200 Ensemble-SWAN due to its slightly superior performance. First, in Figure 7(a), we observe that the accuracy of the Ensemble-SWAN (80.5%) is lower than its corresponding convergence level achieved by the regular SWAN (89.7%). This shows that the ensemble learning approach can only partly compensate for the information lost by undersampling. Similarly, Figure 7(c) shows that the weighted F1-Score of Ensemble-SWAN (85.8%) is lower than the converged F1-Score of the SWAN (91.7%). Conversely, in Figure 7(d) it is evident that the Ensemble-SWAN (82.7%) significantly outperforms the regular SWAN (77.8%) in terms of AUC-ROC. This might be attributed to the fact that undersampling approach creates more diverse subsets than the oversampling approach, which leads to more variability in the model's decision boundaries. This potentially allows the model to better discriminate between converting and non-converting sequences. In Figure 7(b) it is evident that both Ensemble-SWAN and SWAN converge at 94.7% weighted precision.

Another interesting finding is the fact that we observe in Figure 7 that increasing the amount of sub-NNs in the Ensemble-SWAN only minimally increases performance, which might imply that using $N = 50$ sub-NNs is sufficient.

5.2 Uncertainty Analysis

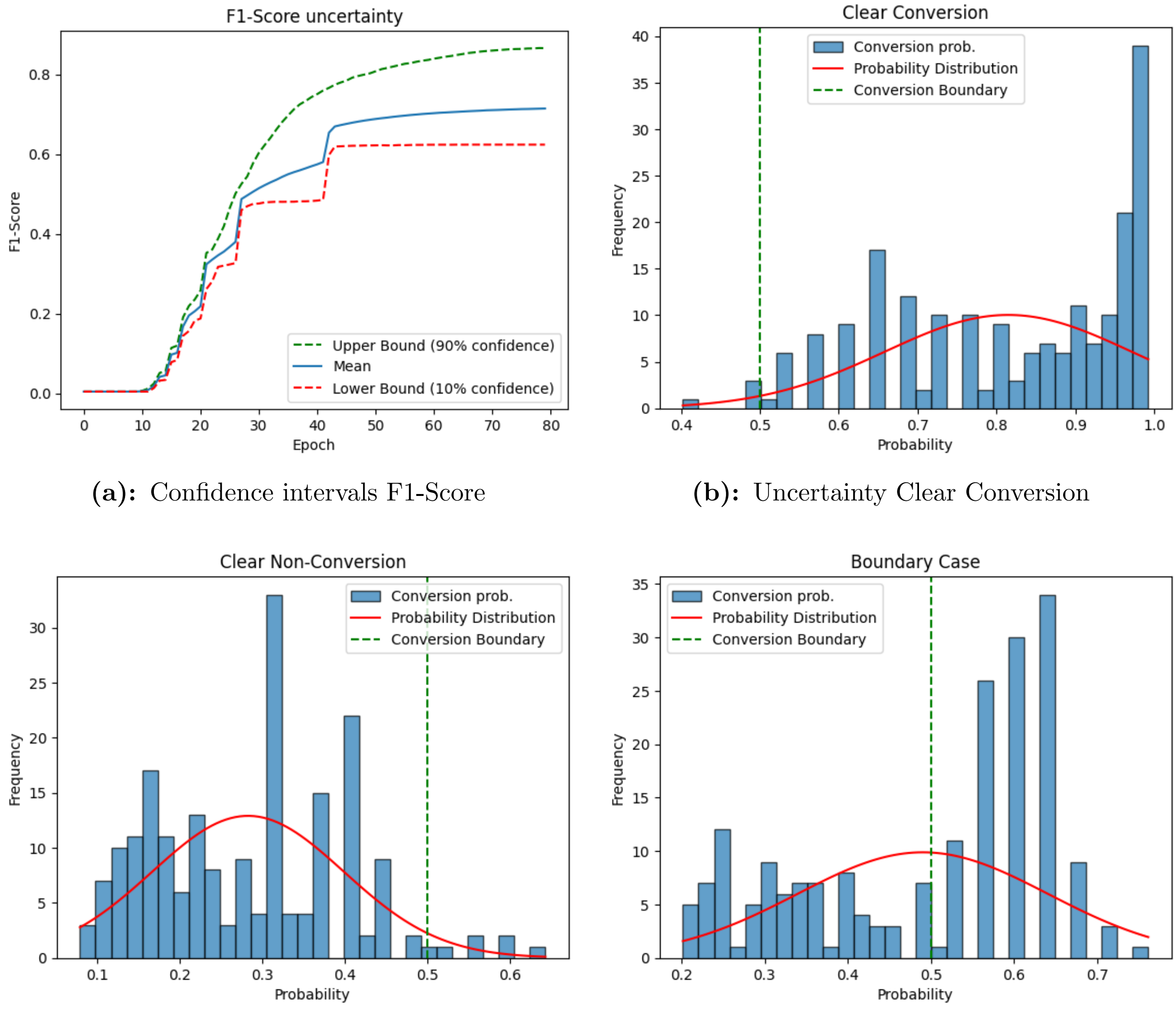
In this Section, we discuss and visualize the results of the two epistemic UQ approaches: the Ensemble approach in Section 5.2.1 and the Monte Carlo Dropout approach in Section 5.2.2.

5.2.1 Uncertainty: Ensemble Approach

In this Section, we will assess the uncertainty regarding the conversion probability predictions for the sub-NNs of the Ensemble-SWAN for $N = 200$ as discussed in Section 4.3.1. The results are summarized in Figure 8. Figure 8(a) shows the mean, upper bound, and lower bound, of the F1-Score when the performance of the individual sub-NNs is evaluated before aggregating the results. The upper and lower bound are calculated using the 90% and 10% confidence intervals, respectively. Evidently, the sub-NNs individually exhibit significantly lower performance than when aggregated in the Ensemble-SWAN. The mean, weighted F1-Score of the $N = 200$, sub-NNs, is 72.1%, whereas when aggregating the conversion predictions before evaluation the model achieves a weighted F1-Score of 85.5% (Ensemble-SWAN). The Ensemble-SWAN weighted F1-Score is significantly higher than the upper bound of the individually evaluated sub-NNs, confirming that the ensemble learning approach substantially enhances model performance.

In Figures 8(b) - 8(d) we visualize the distributions of the conversion predictions of the $N = 200$ sub-NNs before aggregating, for a clear converting sequence, a clear non-converting sequence, and a boundary case, respectively. Please note that the green, vertical, dashed line is the conversion probability threshold of 0.5. We identified a clear converting sequence and a clear non-converting sequence by searching for sequences with a relatively high and low mean prediction conversion probability. The boundary case was identified by searching for a mean prediction conversion probability close to the conversion boundary of 0.5. These Figures show that the predicted conversion probability is quite heavily subject to randomness in the undersampled dataset on which the sub-NNs are trained. For clear converting and non-converting sequences this does not cause any problems as most sub-NNs still correctly classify the sequence. However, for boundary cases as the one depicted in Figure 8(d), the different sub-NNs clearly conclude opposing classifications.

Based on the presented Figures, it is evident that there is considerable amount of uncertainty withing the individual sub-NNs, reaffirming the benefits of an ensemble learning approach. For every input sequence, one could quantify the uncertainty by calculating the variance or standard error of the predicted probabilities. This could especially be useful in various real-world applications where uncertainty aware-decisions are key such as medical image recognition. For example, by calculating the uncertainty for every input token, one could flag boundary cases as depicted in Figure 8(d) for inspection by human eyes.



(a): Confidence intervals F1-Score

(b): Uncertainty Clear Conversion

(c): Uncertainty Clear Non-Conversion

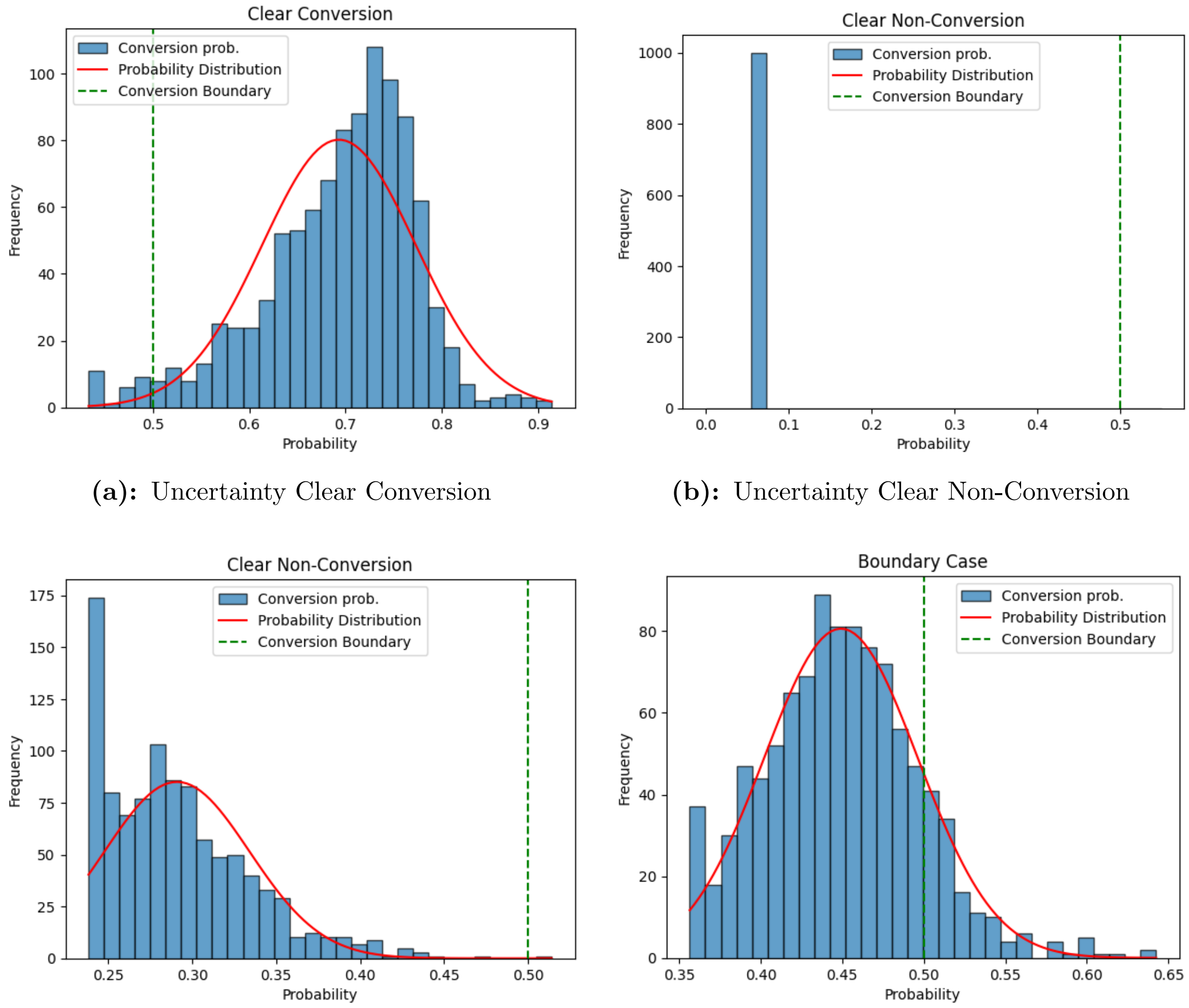
(d): Uncertainty Boundary Case

Figure 8: Epistemic uncertainty in the Ensemble-SWAN visualized by distribution of predictions of sub-NNs.

5.2.2 Uncertainty: Monte Carlo Dropout

As discussed in Section 4.3.2, we can visualize and quantify epistemic uncertainty by using Monte Carlo Dropout. We train the neural network on a dataset which is balanced by random undersampling just like the datasets on which the Ensemble-SWAN is trained. Essentially, the analyzed model can be regarded as one of the sub-NNs from the Ensemble-SWAN. Then, in the test phase, we pass the test data through the neural network $n_{MCD} = 1000$ times in a forward pass with dropout activated.

In Figures 9(a) - 9(d) the distributions of conversion predictions are visualized for a clear converting sequence, two clear non-converting sequences, and a boundary case, respectively. From the figures, we conclude that randomly dropping a quarter of the nodes during the forward pass in the testing phase, quite heavily influences the model's conversion predictions. However, please note that for some input sequences like the one presented in Figure 9(c), no uncertainty is present at all. For most input sequences, there is definitely some epistemic uncertainty associated with the prediction. This epistemic uncertainty might also arise from the fact that the neural



(a): Uncertainty Clear Conversion

(b): Uncertainty Clear Non-Conversion

(c): Uncertainty Clear Non-Conversion 2

(d): Uncertainty Boundary Case

Figure 9: Epistemic uncertainty in one of Ensemble-SWAN's sub-NNs, visualized using Monte Carlo Dropout

network is trained on a relatively small undersampled dataset. Performing the same uncertainty analysis on a similar neural network which is trained on a higher amount of input sequences, might significantly reduce uncertainty present in the model. Again, one could quantify the uncertainty by calculating the variance or standard error of the predicted probabilities, which is useful in situations where uncertainty-aware decisions are key.

For completeness, we also visualized the epistemic uncertainty present in the SWAN, where we balance the data by using oversampling technique ADASYN. These results are shown in Appendix A Figure 10. For most observations, we observe similar levels of uncertainty as we observed in the Ensemble-SWAN Monte Carlo Dropout analysis. However, we have identified some clear non-conversion instances, such as the one depicted in Figure 10(c), where the SWAN produces a small portion of predictions that significantly deviate from the ground truth. This discrepancy is likely to be attributed to training the model on fictitious data, emphasizing the risk of oversampling. For some clear conversion we see slightly less spread in the predictions, which might be attributed to the fact that the SWAN is trained on a dataset 20 times larger than the size of a sub-NN of Ensemble-SWAN.

6 Conclusion

In this paper, we propose a machine learning architecture that aims to solve MTA problems efficiently. The Stacked Web of Attentional Neurons (SWAN) learns to assign conversion credit to individual touchpoints in the input sequence through a simple feed-forward neural network using attention mechanisms. The SWAN's simple architecture enables interpretability and computational efficiency, making it a highly suitable choice for real-world applications. Moreover, we propose an ensemble learning approach, which we refer to as the Ensemble-SWAN, which accounts for the class imbalance problem present in online marketing user journey data. Additionally, in response to the increasing demand for robust UQ methods, we use two techniques to visualize and quantify the epistemic uncertainty in the Ensemble-SWAN.

To evaluate the performance of SWAN and Ensemble-SWAN we trained and tested both models on the Criteo dataset, which is a benchmark dataset in the field of MTA. We compared the results of the SWAN with another state-of-the-art benchmark solution referred to as ARNN, as proposed in Ren et al. (2018). In order to compare to the ARNN, we used oversampling technique ADASYN to balance the training set. In this study, the SWAN demonstrates similar performance in terms of accuracy, precision, F1-Score, and AUC-ROC, despite its simpler architecture. Additionally, we provide insights in the inner workings of the SWAN. The simplicity of the model, allows us to trace back attention weights to individual tokens in the input sequence. Hence, allowing us to assign conversion credit to individual touchpoints in the user journey. In combination with the short training time of a few minutes, this makes the SWAN a highly suitable model for real-world applications.

To account for the class imbalance problem present in online marketing user journey data we introduced the Ensemble-SWAN: an ensemble learning approach leveraging random undersampling. We evaluated the performance of the Ensemble-SWAN for varying number of sub-NNs ($N = 50$, 100 & 200) against the performance of the SWAN. Here, we see that the Ensemble-SWAN demonstrates an approximate 5-10% lower performance than the SWAN in terms of accuracy, precision, and F1-Score. Conversely, the Ensemble-SWAN outperforms the SWAN in terms of AUC-ROC, meaning that the Ensemble-SWAN has more discriminatory power in classification. Unsurprisingly, the lost information due to random undersampling comes at the cost of performance. However, we can conclude that the Ensemble-SWAN can partly compensate for the information loss. Considering that oversampling approaches involve training on fictitious data instances, which may lead to significant failures in real-world applications, accepting a concession of 5-10% in performance could actually prove to be worthwhile.

We also provide two methods to visualize and quantify epistemic uncertainty present in the Ensemble-SWAN. First, we assess the uncertainty regarding the conversion probability predictions for the sub-NNs of the Ensemble-SWAN for $N = 200$. Here we see that predictions can differ significantly when relatively small undersampled datasets are used. Hence, using an ensemble learning approach, like Ensemble-SWAN, is crucial in order to robustify predictions. Next to that, we provide explanation how quantification can be useful in settings where uncertainty aware-decisions are necessary, such as in the field of medical image recognition. Second, Monte

Carlo Dropout can be used to visualize and quantify uncertainty of individual input sequences. Monte Carlo Dropout disables a portion of the nodes in the neural network during inference to estimate uncertainty. Here, we illustrate that the model exhibits uncertainty for certain input sequences, while displaying no uncertainty for other input sequences. In some settings, it would be useful to flag the inputs for which the model is uncertain, to be further evaluated with human eyes. To conclude, both approaches show significant epistemic uncertainty present in the Ensemble-SWAN, emphasizing the need for caution in real-world applications.

All things considered, the SWAN offers a highly suitable solution for addressing MTA problems in real-world applications, thanks to its interpretability and computational efficiency. Moreover, the Ensemble-SWAN effectively handles the class imbalance problem using ensemble learning techniques and random undersampling, while minimizing performance degradation caused by the information loss. However, one should acknowledge and be aware of the presence of epistemic uncertainty in deep learning models, such as the Ensemble-SWAN, when implemented in practice.

For further research the possibilities are manifold. First, it would be interesting to evaluate the performance of the SWAN in other domains where interpretability and computational feasibility are important, such as credit card fraud detection and DNA sequence classification. Moreover, one could combine different types of neural network architectures in one ensemble learning approach to investigate if this enhances performance in the MTA domain. In this research, the Ensemble-SWAN consists of exactly similar architectures trained on different datasets rather than different models. Another interesting field to explore further is UQ. In this research we address the epistemic uncertainty of the model, which can be reduced by increasing the amount of data and improving model complexity. However, aleatoric uncertainty, which is irreducible, is not addressed in this research. It would be interesting to develop a granular UQ technique that integrates both epistemic and aleatoric uncertainty in a single measure.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., … others (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, *76*, 243–297.

Arava, S. K., Dong, C., Yan, Z., Pani, A. et al. (2018). Deep neural net with attention for multi-channel multi-touch attribution. *arXiv preprint arXiv:1809.02230*.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*, 123–140.

Buhalis, D. & Volchek, K. (2021). Bridging marketing theory and big data analytics: The taxonomy of marketing attribution. *International Journal of Information Management*, *56*, 102253.

Bunkhumpornpat, C., Sinapiromsaran, K. & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Advances in knowledge discovery and data mining: 13th pacific-asia conference, pakdd 2009 bangkok, thailand, april 27-30, 2009 proceedings 13* (pp. 475–482).

Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y. & Guo, D. (2017). Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the tenth acm international conference on web search and data mining* (pp. 661–670).

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Dalessandro, B., Perlich, C., Stitelman, O. & Provost, F. (2012). Causally motivated attribution for online advertising. In *Proceedings of the sixth international workshop on data mining for online advertising and internet economy* (pp. 1–9).

Davis, J. & Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on machine learning* (pp. 233–240).

De Haan, E., Kannan, P., Verhoef, P. C. & Wiesel, T. (2015). The role of mobile devices in the online customer journey. *MSI Marketing Science Institute*, 15–124.

Diemert, E., Meynet, J., Galland, P. & Lefortier, D. (2017). Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the adkdd'17* (pp. 1–6).

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems: First international workshop, mcs 2000 cagliari, italy, june 21–23, 2000 proceedings 1* (pp. 1–15).

Egele, R., Maulik, R., Raghavan, K., Lusch, B., Guyon, I. & Balaprakash, P. (2022). Autodeuq: Automated deep ensemble with uncertainty quantification. In *2022 26th international conference on pattern recognition (icpr)* (pp. 1908–1914).

Fort, S., Hu, H. & Lakshminarayanan, B. (2019). Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*.

Fu, J., Zheng, H. & Mei, T. (2017). Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4438–4446).

Gal, Y. & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050–1059).

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, *57*, 345–420.

Goldfarb, A. & Tucker, C. (2011). Online display advertising: Targeting and obtrusiveness. *Marketing Science*, *30*(3), 389–404.

Han, H., Wang, W.-Y. & Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *Advances in intelligent computing: International conference on intelligent computing, icic 2005, hefei, china, august 23-26, 2005, proceedings, part i 1* (pp. 878–887).

He, H., Bai, Y., Garcia, E. A. & Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)* (pp. 1322–1328).

Hüllermeier, E. & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, *110*, 457–506.

Izmailov, P., Vikram, S., Hoffman, M. D. & Wilson, A. G. G. (2021). What are bayesian neural network posteriors really like? In *International conference on machine learning* (pp. 4629–4640).

Japkowicz, N. & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, *6*(5), 429–449.

Ji, W., Wang, X. & Zhang, D. (2016). A probabilistic multi-touch attribution model for online advertising. In *Proceedings of the 25th acm international on conference on information and knowledge management* (pp. 1373–1382).

Jiang, H., Kim, B., Guan, M. & Gupta, M. (2018). To trust or not to trust a classifier. *Advances in neural information processing systems*, *31*.

Kannan, P., Reinartz, W. & Verhoef, P. C. (2016). *The path to purchase and attribution modeling: Introduction to special section* (Vol. 33) (No. 3). Elsevier.

Kumar, S., Gupta, G., Prasad, R., Chatterjee, A., Vig, L. & Shroff, G. (2020). Camta: Causal attention model for multi-touch attribution. In *2020 international conference on data mining workshops (icdmw)* (pp. 79–86).

Lemaître, G., Nogueira, F. & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, *18*(1), 559–563.

Li, Z., Cheng, W., Chen, Y., Chen, H. & Wang, W. (2020). Interpretable click-through rate prediction through hierarchical attention. In *Proceedings of the 13th international conference on web search and data mining* (pp. 313–321).

Liu, X.-Y., Wu, J. & Zhou, Z.-H. (2008). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(2), 539–550.

Moffett, T., Pilecki, M. & McAdams, R. (2014). The forrester wave: Cross-channel attribution providers, q4 2014. *Forrester Research Inc., Cambridge, MA (https://services. google. com/fh/files/misc/forrester_cca_wave_q42014. pdf)*.

Pascanu, R., Mikolov, T. & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310–1318).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … others (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y. & Wang, J. (2016). Product-based neural networks for user response prediction. In *2016 ieee 16th international conference on data mining (icdm)* (pp. 1149–1154).

Raffel, C. & Ellis, D. P. (2015). Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*.

Raftery, A. E., Madigan, D. & Hoeting, J. A. (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, *92*(437), 179–191.

Rao, K. D., Kushwaha, H., Verma, A. K. & Srividya, A. (2007). Quantification of epistemic and aleatory uncertainties in level-1 probabilistic safety assessment studies. *Reliability Engineering & System Safety*, *92*(7), 947–956.

Rasmussen, C. E., Williams, C. K. et al. (2006). *Gaussian processes for machine learning* (Vol. 1). Springer.

Ren, K., Fang, Y., Zhang, W., Liu, S., Li, J., Zhang, Y., … Wang, J. (2018). Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th acm international conference on information and knowledge management* (pp. 1433–1442).

Shao, X. & Li, L. (2011). Data-driven multi-touch attribution models. In *Proceedings of the 17th acm sigkdd international conference on knowledge discovery and data mining* (pp. 258–264).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929–1958.

Tarawneh, A. S., Hassanat, A. B., Altarawneh, G. A. & Almuhaimeed, A. (2022). Stop oversampling for class imbalance learning: A review. *IEEE Access*, *10*, 47643–47660.

Tomek, I. (1976). A generalization of the k-nn rule. *IEEE Transactions on Systems, Man, and Cybernetics*(2), 121–126.

Vallabhajosyula, S., Sistla, V. & Kolli, V. K. K. (2022). Transfer learning-based deep ensemble neural network for plant leaf disease detection. *Journal of Plant Diseases and Protection*, *129*(3), 545–558.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wen, Y., Tran, D. & Ba, J. (2020). Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*.

Wooff, D. A. & Anderson, J. M. (2015). Time-weighted multi-touch attribution and channel relevance in the customer journey to online purchase. *Journal of statistical theory and practice*, *9*, 227–249.

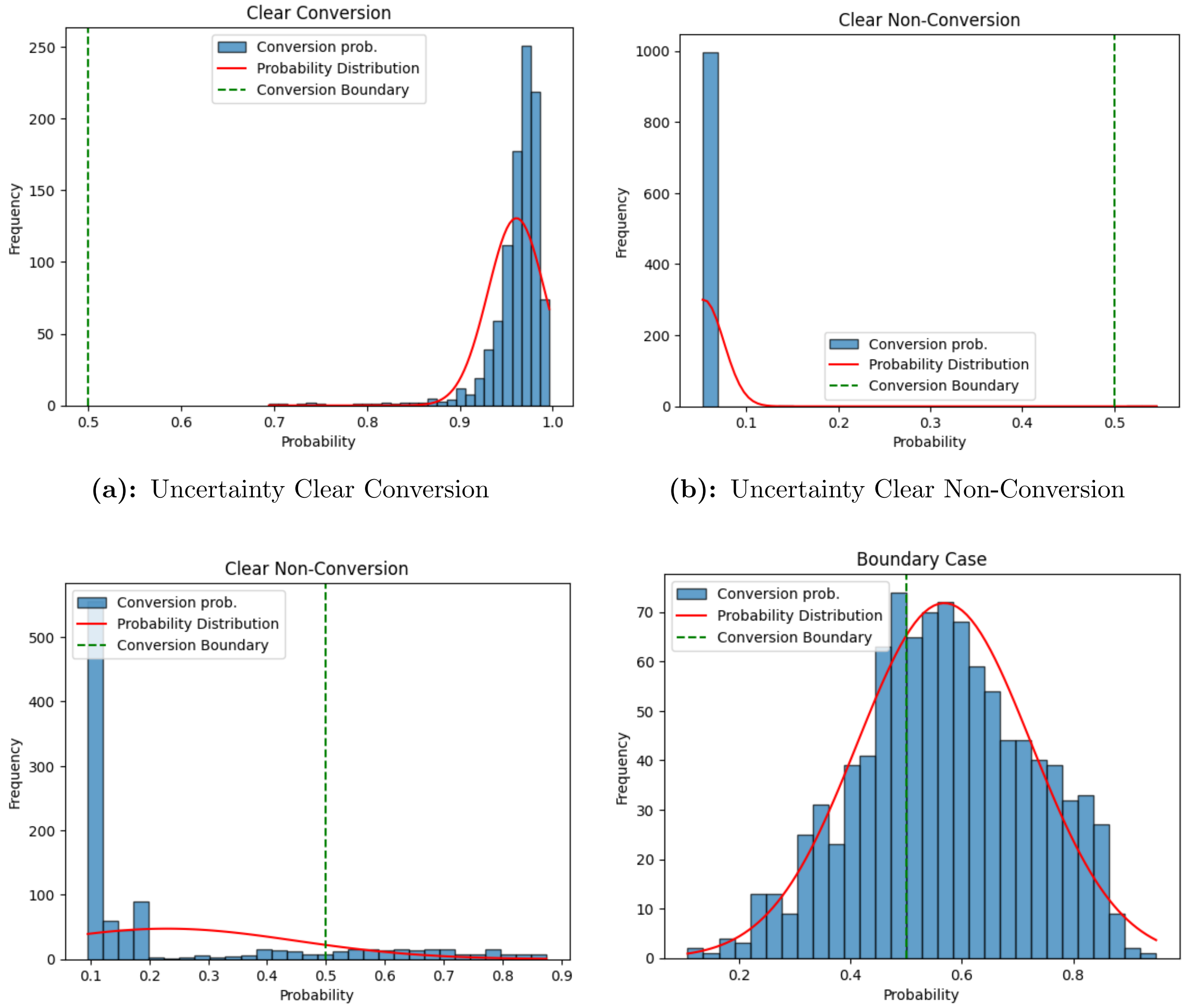
Xu, J., Shao, X., Ma, J., Lee, K.-c., Qi, H. & Lu, Q. (2016). Lift-based bidding in ad selection. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 30).

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S. J. & Kumar, S. (2019). Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*.

Zhang, Y., Wei, Y. & Ren, J. (2014). Multi-touch attribution in online advertising with survival theory. In *2014 ieee international conference on data mining* (pp. 687–696).

Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., … Gai, K. (2019). Deep interest evolution network for click-through rate prediction. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 5941–5948).

A Appendix



(a): Uncertainty Clear Conversion

(b): Uncertainty Clear Non-Conversion

(c): Uncertainty Clear Non-Conversion 2

(d): Uncertainty Boundary Case

Figure 10: Epistemic uncertainty in 'regular' SWAN (oversampled data) visualized using Monte Carlo Dropout.

B Code Overview

Please find a brief description of all the Python scripts used in this research below. Note that some of the scripts are in Jupyter notebook format and some are in regular Python format. For computational reasons some scripts are run on a more powerful computer, so for those scripts we do not have the output in Jupyter notebook format. All scripts can be found at `https://github.com/pimweterman/Ensemble-SWAN.git`.

Data_Preprocessing.ipynb. contains the necessary data pre-processing steps such as obtaining all sequences with minimum length 3 and maximum length 20.

Descriptive_Statistics_Criteo.ipynb contains the code for the descriptive statistics in Section 3.

ARRN_replication.py contains the code of the ARNN as proposed by (Ren et al., 2018), which is used to benchmark the performance of the SWAN.

SWAN.ipynb contains the code of the Stacked Web of Attentional Neurons (SWAN). This file also contains the code for obtaining the oversampled dataset using ADASYN.

Ensemble-SWAN.py contains the code of the Ensemble-SWAN. This file also contains the code for obtaining the undersampled datasets. Additionally, this file contains the code for the ensemble uncertainty quantification approach.

UQ_MCD_Undersampled.ipynb contains the code for the Monte Carlo Dropout uncertainty quantification for one of the (undersampled) sub-NNs of the Ensemble-SWAN.

UQ_MCD_Oversampled.ipynb contains the code for the Monte Carlo Dropout uncertainty quantification for the (oversampled) SWAN.

Plots_SWAN_Ensemble-SWAN.ipynb contains the code for all the Figures in Section 5.