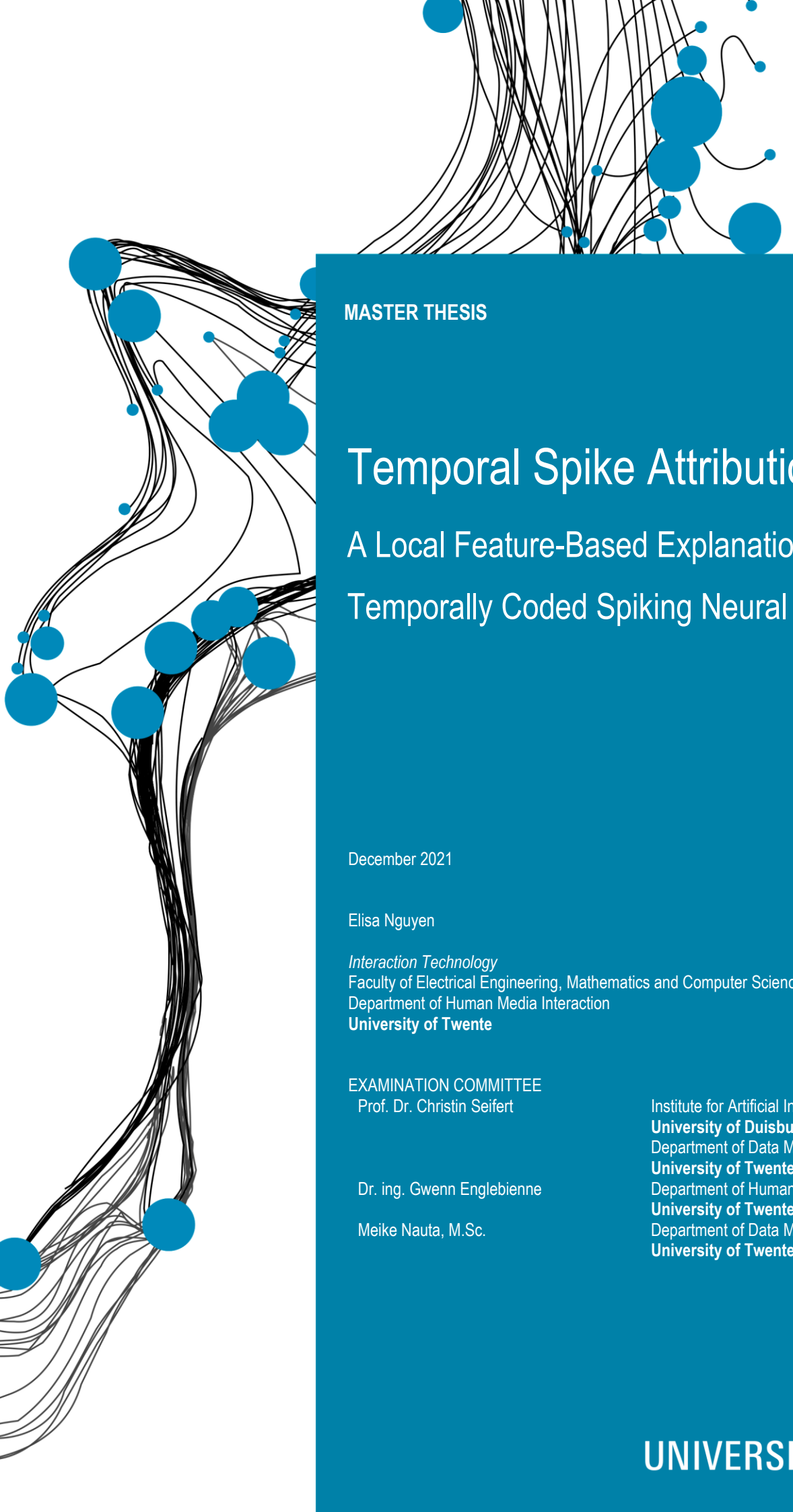


MASTER THESIS

Temporal Spike Attribution

A Local Feature-Based Explanation for Temporally Coded Spiking Neural Networks

December 2021

Elisa Nguyen

Interaction Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Department of Human Media Interaction
University of Twente

EXAMINATION COMMITTEE

Prof. Dr. Christin Seifert	Institute for Artificial Intelligence in Medicine, **University of Duisburg-Essen**
	Department of Data Management & Biometrics **University of Twente**
Dr. ing. Gwenn Englebienne	Department of Human Media Interaction **University of Twente**
Meike Nauta, M.Sc.	Department of Data Management & Biometrics **University of Twente**

UNIVERSITY OF TWENTE.

ABSTRACT

Machine learning algorithms are omnipresent in today's world. They influence what movie one might watch next or which advertisements a person sees. Moreover, AI research is concerned with high-stakes application areas, such as autonomous cars or medical diagnoses. These domains pose specific requirements due to their high-risk nature: In addition to predictive accuracy, models have to be transparent and ensure that their decisions are not discriminating or biased. The definition of performance of artificial intelligence is therefore increasingly extended to requirements of transparency and model interpretability. The field of Interpretable Machine Learning and Explainable Artificial Intelligence concerns methods and models that provide explanations for black-box models.

Spiking neural networks (SNN) are the third generation of neural networks and therefore also black-box models. Instead of real-valued computations, SNNs work with analogue signals and generate spikes to transmit information. They are biologically more plausible than current artificial neural networks (ANN) and can inherently process spatio-temporal information. Due to their ability to be directly implemented in hardware, their implementation is more energy-efficient than ANNs. Even though it has been shown that SNNs are as powerful, they have not surpassed ANNs so far. The research community is largely focused on optimising SNNs, while topics related to interpretability and explainability in SNNs are rather unexplored.

This research contributes to the field of Explainable AI and SNNs by presenting a novel local feature-based explanation method for spiking neural networks called Temporal Spike Attribution (TSA). TSA combines information from model-internal state variables specific to temporally coded SNNs in an addition and multiplication approach to arrive at a feature attribution formula in two variants, considering only spikes (TSA-S) and also considering non-spikes (TSA-NS). TSA is demonstrated on an openly-available time series classification task with SNNs of different depths and evaluated quantitatively with regard to faithfulness, attribution sufficiency, stability and certainty. Additionally, a user study is conducted to verify the human-comprehensibility of TSA. The results validate TSA explanations as faithful, sufficient, and stable. While TSA-S explanations are more stable, TSA-NS explanations are superior in faithfulness and sufficiency, which suggests relevant information for the model prediction to be in the absence of spikes. Certainty is provided in both variants, and the TSA-S explanations are largely human-comprehensible where the clarity of the explanation is linked to the coherence of the model prediction. TSA-NS, however, seems to assign too much attribution to non-spiking input, leading to incoherent explanations.

ACKNOWLEDGEMENTS

This thesis concludes my studies and master research at the University of Twente. It was a time full of learning, interesting topics and great memories. Many people helped me throughout this thesis whom I would like to express my gratitude for.

First of all, I would like to thank my supervisors, Prof. Dr. Christin Seifert, Dr. ing. Gwenn Englebienne, and Meike Nauta, M.Sc., for the constant support, interesting discussions and active involvement in this project. Your guidance attributed greatly to the quality of this work and I want to thank you for your time and dedication. I highly appreciated your quick thinking and ideas to overcome the challenges along the way. Thank you for trusting me with a topic I was rather unfamiliar with at first, your patience during all the meetings that went overtime, and seeing potential in my abilities. Specifically, thank you to Christin and Meike, for the initial idea for this research which I enjoyed studying very much and for allowing me to access the research infrastructure at UK Essen.

Secondly, I would like to express my gratitude for everyone who helped me with the practical part of the thesis: Jörg Schlötterer for organising my access to the high performance computing cluster at the Institute of Artificial Intelligence in Medicine at UK Essen, where most of the experiments of this thesis were run. Thank you to Kata, Kevin and Rinalds for your time and efforts to both pilot the user study and perform the cluster analysis for the qualitative evaluation. Thank you to everyone who participated in the survey and enabled this research. Moreover, I would like to mention a thanks to Dr. Friedemann Zenke and his team for providing tutorials and open access code on building and training SNNs with surrogate gradient learning. It helped me incredibly with the use case implementation.

A special thanks to Overleaf and Google Cloud, without which all my progress would probably have been lost when my laptop broke this summer. Also thank you to Cas, for saving all the data from the broken laptop, so that nothing was lost in the end.

Lastly, I would like to express some personal thanks to my family and friends who shared this exciting journey with me. Unfortunately there is not enough space to name everyone, therefore I only mention a few that contributed particularly to this thesis. Cám ơn ba má, my parents, who sparked my passion for learning and curiosity, sent me care-packages and recipes when I missed home. Thank you to Michael, for many fruitful discussions, all the proofreading, and sharing excitement for the research process. Thank you Sanjeet, for motivating me when I needed it and always willing to go through calculations with me. Thank you to my favourite library partners who made working on a solo-project like the thesis less lonely: Daphne, Domi, Oscar, Robi, Umbi. Finally, thank you Simon, for selflessly pushing me to follow my passion and dreams, no matter where they may lead me.

“Whenever an AI system has a significant impact on people’s lives, it should be possible to demand a suitable explanation of the AI system’s decision-making process.”

High-Level Expert Group on AI of the European Commission in: Ethics guidelines for trustworthy AI (2018)

“When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.”

Donald O. Hebb (1904 - 1985) in: The organization of behaviour

In other words: *“Neurons that fire together, wire together.”*

Hebb’s Law

TABLE OF CONTENTS

List of Abbreviations **xii**

1 Introduction **1**
1.1 Problem Statement and Research Questions . . . 3
1.2 Outline . . . 4

2 Background **5**
2.1 Foundations of Spiking Neural Networks . . . 5
2.1.1 Neural Networks and their Biological Inspiration . . . 5
2.1.2 Spiking Neuron Model . . . 6
2.1.3 Neural Code . . . 8
2.1.4 Learning algorithm . . . 9
2.1.5 Building Spiking Neural Networks for Research . . . 9
2.2 Foundations of Explainable Artificial Intelligence . . . 11
2.2.1 Terminology . . . 11
2.2.2 Taxonomies . . . 11

3 Related Work **14**
3.1 Interpretable Spiking Neural Networks . . . 14
3.1.1 Global Interpretability through Feature Strength Functions . . . 14
3.1.2 Local Explanations with the Spike Activation Map . . . 17
3.2 Explainable AI with Time Series . . . 18
3.2.1 Explanation Methods with Time Series . . . 18
3.2.2 Desired Properties of Explanations for Time Series . . . 20
3.3 Common Architectures of Spiking Neural Networks . . . 21

4 Use Case Model and Data **23**
4.1 Data . . . 23
4.1.1 Dataset Description . . . 23
4.1.2 Data Preprocessing . . . 25
4.2 Models . . . 25

4.2.1 SNN Architecture Choices . . . 25
4.2.2 Model Development . . . 26
4.2.3 Final Models . . . 28

5 Temporal Spike Attribution Explanations 29
5.1 Feature Attribution Definition . . . 29
5.2 Temporal Spike Attribution Components . . . 30
5.2.1 Influence of Spike Times . . . 30
5.2.2 Influence of Model Parameters . . . 31
5.2.3 Influence of the Output Layer's Membrane Potential . . . 32
5.3 Temporal Spike Attribution Formula . . . 32
5.4 Visualisation . . . 35
5.4.1 First Iteration - Initial Visualisation . . . 35
5.4.2 Second Iteration - Spikes and Colours . . . 37
5.4.3 Third Iteration - Confidence . . . 37

6 Explanation Qualities 39
6.1 Technical evaluation . . . 40
6.1.1 Experimental Setup . . . 40
6.1.2 Results and Discussion . . . 46
6.2 User evaluation . . . 55
6.2.1 User Study Design . . . 55
6.2.2 Results and Discussion . . . 57
6.3 Implication and Outlook . . . 62

7 Discussion 64
7.1 Answer to Research Questions . . . 64
7.2 Reflection on Evaluation Framework . . . 66
7.3 Reflection on Explaining Deep Models . . . 67
7.4 Reflection on Non-Spiking Attribution . . . 67
7.5 Limitations . . . 68

8 Conclusion and Future Work 71

References xiii

A Overview of related work in SNN research xx

B Supplementary material about model development xxv

C Examples from the quantitative analysis xxvii

D Unmasked simulation explanations for user study **xxxii**

E Clustering Instructions **xxxiv**

F Results of Inductive Cluster Analysis **xxxv**

Attachment **xxxvii**

LIST OF ABBREVIATIONS

ADL	Activities of Daily Living Recognition using Binary Sensors
AI	Artificial Intelligence
ANN	Artificial Neural Network
CI	Confidence Interval
EEG	Electroencephalogram
FS	Feature Segment
FSF	Feature Strength Function
GDPR	General Data Protection Regulation
IML	Interpretable Machine Learning
LIF	Leaky Integrate-and-Fire
MTS	Multivariate Time Series
NCS	Neuronal Contribution Score
PSP	Postsynaptic Potential
SAM	Spike Activation Map
SEFRON	Synaptic Efficacy Function-based leaky integrate-and-fire neuRON
SNN	Spiking Neural Network
STDP	Spike-Time Dependent Plasticity
TSA	Temporal Spike Attribution
TSA-S	Temporal Spike Attribution - Only Spikes
TSA-NS	Temporal Spike Attribution - Non-Spikes included
TSCS	Temporal Spike Contribution Score
TTFS	Time To First Spike
UCI	University of California, Irvine
VSLI	Very Large-Scale Integration
XAI	eXplainable Artificial Intelligence

1 INTRODUCTION

The use of artificial intelligence and machine learning in real-life applications is common in the year 2021. The areas of application are wide, ranging from private use, e.g. recommendation systems like Netflix [1] to decisions that have a larger impact on the individual, such as credit scoring applications [2] or aid in medical diagnosis [3], to name a few. While private use scenarios like Netflix recommendations are already in practice, there are general inhibitions for practical deployment of safety or ethically critical AI applications such as medical diagnosis. In these cases, it is not only important for a model to have high predictive performance, but also to understand why an algorithm arrived at a certain prediction [4]. The decision must be transparent to a certain degree to ensure that the algorithm makes a prediction based on criteria that make sense and are not based on discriminating factors [5]. Interpretable Machine Learning (IML) and eXplainable Artificial Intelligence (XAI) are fields of research that are concerned with this problem [6]. The methods developed in these fields aim at providing transparency to different degrees and target groups in order to foster trust in machine learning applications. This is important for critical fields, in which a faulty decision could have major consequences [7].

While simple models like linear regression, or rule-based systems like decision trees are considered intrinsically interpretable, Artificial Neural Networks (ANN) uncover non-linearities in data and make use of these for their predictions. Consequently, their decision behaviour becomes a black box for humans. As these models reached high predictive performances for complex problems like image classification, they are often applied to the above-mentioned critical areas. Beyond the general motivation to provide transparency and encourage trust in machine learning applications, the relevance of interpretability is also highlighted through recent ethical guidelines like the European Commission's ethics guidelines for trustworthy AI [8] where transparency "including traceability, explainability and communication" [8, p. 14] of AI systems is named as one of seven key requirements, and recent legislation like the General Data Protection Regulation (GDPR). The GDPR was introduced in the European Union in 2018 [9] and emphasises trustability, transparency, and fairness of machine learning algorithms. Thus, there is a strong motivation for research in IML and XAI from a social, ethical and legal point of view.

As a result of the expanded research interest in IML and XAI, the performance definition of machine learning models is increasingly extended from mainly predictive accuracy to model interpretability [6], which underlines the importance of this field further. Model interpretability, however, has no standard evaluation practice so far. The main reason is the high diversity in explanation methods that provide interpretability, which differ in scope, applicability and objective [10]. Therefore, any work that studies an explanation method should also study its evaluation criteria, based on the use case, to provide a reliable interpretability assessment of the model. In this work, a novel explanation method is presented, including an evaluation criteria analysis and evaluation on a specific use case to assess the explanatory performance of the method.

One type of black-box models are neural networks. Neural networks are based on their computational units, called neurons. Based on the neurons, three generations of neural networks can be distinguished. The first generation operates with McCulloch-Pitts neurons, which are

threshold gates. The second generation uses activation functions for computation, which can be non-linear and thus uncover non-linearities in the data. Both of these fall in the category of ANNs, which is mostly understood under the term neural network. Spiking Neural Networks (SNN) are less well-known. They are the third generation of neural networks and apply spiking neurons as computational units [11]. Spiking neurons emit pulses at certain times, similar to a biological neuron, to transmit information. Therefore, SNNs use spatio-temporal information of the timing of a pulse as well as the frequency of pulses in their computation. By their ability to use pulse timing, they are biologically more plausible than their predecessors. Furthermore, SNNs yield the potential to be implemented into analogue Very Large-Scale Integration (VSLI) hardware, which is energy efficient and space-saving [12], so that SNNs can run at lower energy cost than current ANNs.

It has been shown that SNNs are at least as powerful as the second-generation ANNs [11]. However, there is no current state-of-the-art SNN learning algorithm yet. Since gradients are undefined for binary pulses, the error backpropagation learning algorithm cannot be applied. As a consequence, SNNs have not achieved significant improvements in terms of predictive performance in comparison to ANNs. Hence, most research in SNNs is focused on the development of a suitable learning algorithm and efficient SNN architecture. Nevertheless, the outlook of more energy-efficient machine learning implementations that are at least as powerful as current ANNs indicates that SNNs will remain subject to future research. Moreover, progress in the research in neuromorphic VSLI hardware may have an accelerating impact on SNN research as well. Due to the SNN's inherent ability to process spatio-temporal data, they are predestined to process sensor data. This makes them suitable for critical domains such as autonomous control and medical diagnosis. For example, a previous study showed the success of SNNs as autonomous controller systems for robots, where the low energy and memory consumption of SNNs are mentioned as large advantages compared to ANNs [13]. A more recent study [14] presented an implementation of SNNs on neuromorphic hardware in autonomous robot control with integration of off-the-shelf and smartphone technology. Azghadi et al. (2020) [15] demonstrate SNNs as a complementary part to ANNs which is dedicated to and more efficient in processing of biomedical signals in healthcare applications at the edge. Moreover, first studies imply stronger adversarial robustness of SNNs in comparison to ANNs especially in black-box attack scenarios thanks to their inherent temporal dynamics [16]. All the above-mentioned points support further research into SNNs, even though they have not yet surpassed second-generation ANNs in predictive performance. Nonetheless, it will be beneficial to already have methods for interpretability in place, so that the implementation of SNNs in productive applications can offer model interpretability at the same time. The requirements of transparency and fairness will likely be asked of SNNs in the same way as of current ANNs. This work aims at contributing to this rather unexplored and novel field of research, and provide a study for the generation of explanations for SNNs.

In detail, the generation of local explanations of SNN models is studied, i.e., the explanation of a certain model prediction outcome. Local explanations show why a particular input leads to the model prediction [10]. They are interesting to study in an unexplored field such as the explainability of SNN models because local explanations highlight the relation between data instances and the model. Therefore, a local explanation method provides information about the model behaviour at instance-level. The investigation of model behaviour at this granular level is interesting for both users and model developers. For the users, a local explanation fulfils the user's legal rights for transparency and explanation regarding algorithms [9]. For model developers, a local explanation provides possibilities to understand SNN modelling with regard to particular data instances. This allows them to identify the reasons for model behaviour that might otherwise not have been found and improve the model if needed. Furthermore, this level of insight into the model might enable discoveries about e.g. SNN behaviour or the data as well [7]. Thus, local explanations, especially for highly variable models such as SNNs that

exhibit many parameters and architectural options, is interesting to study as it facilitates an inspection of the model behaviour at instance-level. It is possible to inspect the effect of SNN's inherent temporal dynamics for example, which is particularly appealing for time series data. In ANNs, the temporal dimension is often encoded in summary statistics of a window of the time series, whereas SNNs do not necessarily require windowing. To the best of the author's knowledge, there exists limited related work for local explanations for SNNs, and no studies into explainability of SNNs on time series data, so that a study in this direction likely provides novel and interesting insights into the explainability of SNNs.

The inherent temporal dynamics of SNNs set them apart from the previous generation neural networks. These are reflected in the SNN model's internal variables. Therefore, it makes sense to develop a local explanation method around these variables to provide an SNN-specific explanation. Such an explanation could capture the effects of spatio-temporal learning and show the behaviour of SNNs. As there is little previous work that such a method could build upon, a novel vanilla feature-attribution based explanation method is targeted which extracts the attributions of input features for a particular output and builds a saliency-type explanation. Future work could then build on this method, to develop other, more complex explanations for SNNs involving causal relationships or counterfactuals, for example.

1.1 Problem Statement and Research Questions

The problem statement for developing a reliable local explanation method for an SNN on a time series classification task can be formulated as follows: Let f be a trained SNN model and $X \in \mathbb{R}^{D \times T}$ the spiking data with D input dimensions and duration of T. The objective is to develop an explanation method $e(f, x, t)$ that shows the attributions of input $x \in X$'s features at time t on the model's output $f(x, t) = \hat{y}$ for x at time t. For this, the model's internal variables, such as the weights W, the spiking behaviour expressed in spike trains S as well as the membrane potentials U are to be used, so that the explanation reflects the model behaviour.

Thus, this research sets out to answer the following research question:

How can the predictions of a temporally coded spiking neural network be explained reliably?

This research question can be broken down into two parts, which cover the development of an explanation method (S-RQ1) and the evaluations and reliability of said explanation (S-RQ2).

1. S-RQ1: *How can feature attribution be calculated for temporally coded spiking neural networks?*
2. S-RQ2: *How can the quality of local feature-attribution-based explanations extracted from SNNs be measured?*

To answer S-RQ1, an SNN model-agnostic algorithm to compute feature attribution based on the respective impacts of W, S and U to the relation between x and $\hat{y}$ is developed through an addition and multiplication approach. A theoretical standpoint is initially chosen, but the method is applied to temporally coded SNNs, which are built and trained on a time series classification use case. These models act as the basis of the work, for both method development as well as evaluation in S-RQ2. The feature attribution algorithm then presents the method that answers S-RQ1.

To answer S-RQ2, desired explanation qualities are deduced from related literature, under consideration of the scope, application, and target group of the explanation method. These are translated into a thorough technical and user evaluation, including concrete metrics and study

design. By applying this evaluation method to the explanations extracted from the underlying SNN models, S-RQ2 is answered while assessing the explanation method from S-RQ1. Thus, both sub-research questions contribute to answering the overarching research question, by setting the framework to develop and assess a local explanation method for temporally coded spiking neural networks.

1.2 Outline

This thesis is structured as follows. First, as neither spiking neural networks nor explainable artificial intelligence is part of the standard curriculum in machine learning, chapter 2 gives an introduction into those topics. Chapter 3 presents existing related work in the field of interpretable SNNs. Additionally, related work concerning SNNs with time series data and XAI methods with time series data is explored to choose a sensible SNN model architecture as well as examine existing XAI work for best practices as a basis for the experimental use case.

In chapter 4, the data, task and architecture of the underlying SNN models at the basis of this research are explained. Afterwards, the first sub-research question is studied by the formal definition of a feature attribution computation in chapter 5. Chapter 6 presents the evaluation qualities and metrics, as well as the experimental results and discussion on an openly available time series dataset. The research questions are answered and the limitations of this work are reflected in chapter 7. In chapter 8, the main points of this thesis are summarised and concluded, as well as an outlook on potential future work given.

2 BACKGROUND

In this chapter, relevant background information and vocabulary from the fields around spiking neural networks as well as explainable artificial intelligence is given to equip the reader with the background knowledge necessary for this thesis.

2.1 Foundations of Spiking Neural Networks

As spiking neural networks are a rather specific type of neural network which is more popular in neuroscience rather than the overall field of machine learning, this section shall give a short introduction to the relevant vocabulary and architecture concepts for this thesis. Spiking neural networks are characterised by several architectural choices, namely the spiking neuron model, the neural code, and the learning algorithm. Furthermore, the implementation possibilities of spiking neural networks for experiments is shortly depicted.

2.1.1 Neural Networks and their Biological Inspiration

Artificial neural networks are modelled after the structures found in the brain, a biological neural network [17]. In the brain, multiple neuron cells[1] are linked to each other through synapses[2]. Neurons exchange information in the form of chemical neurotransmitters, which affect the neuron's membrane potentials. Excitatory (i.e., increase of postsynaptic neuron's membrane potential) and inhibitory (i.e., decrease of postsynaptic neuron's membrane potential) are distinguished. The change in potential can lead a neuron to activate in case of sufficient stimulation. Once activated, a neuron communicates with its downstream neurons by firing an action potential, also called spike, at activation time [18] (Figure 2.1). Directly after spiking, the neuron enters a refractory period, in which spiking is not possible during absolute refractoriness and is less likely during relative refractoriness. After some time, the neuron's membrane potential recovers to the resting state. It is assumed that the information about a stimulus to the brain, e.g. a sound, is contained in the number of spikes and spike timings, which spiking neural networks (SNN) make use of [17].

SNNs are known as the third generation of artificial neural networks [11] (Figure 2.2). Generations are defined based on the computations in the neurons. After the first generation of McCulloch-Pitts neurons, which are threshold gates, and the second generation of artificial neurons with continuous activation functions, SNNs implement spiking neurons and learn spatio-temporal patterns. Spiking neurons emit pulses at certain times, similar to action potentials in biological neurons, to transmit information. Therefore, SNNs are closer to the biological reality [17].

[1] The brain consists of both neuron cells and glia cells. Glia cells are omitted for brevity.

[2] For simplicity, only chemical synapses are referred to when mentioning synapses in this work.

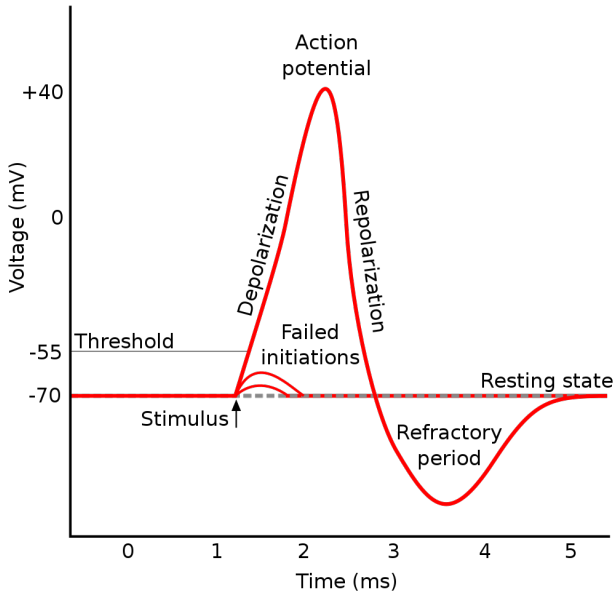


Figure 2.1: Action potential of a neuron[3]

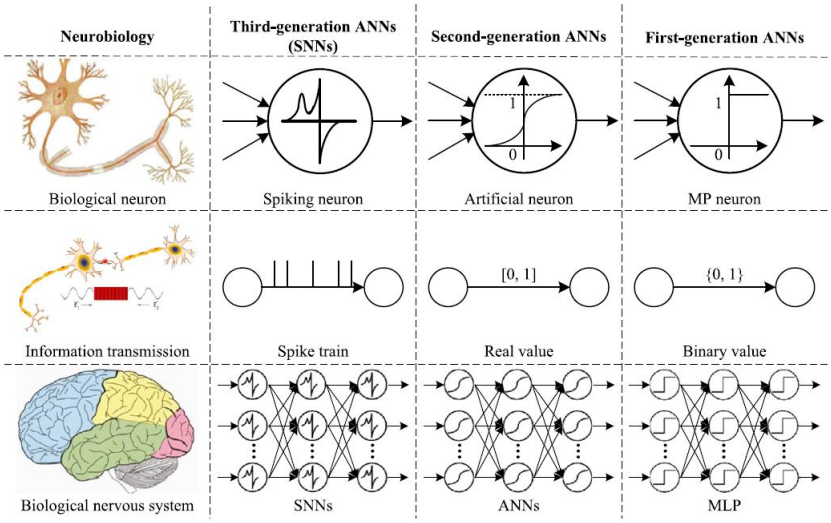


Figure 2.2: Comparison of three generations of neural networks and neurobiology [19, p. 259]

2.1.2 Spiking Neuron Model

Multiple models from the area of neuroscience exist for the definition of spiking neurons. These dictate the temporal dynamics and the spiking behaviour of a neuron. The Hodgkin-Huxley model [18] represents the most biologically accurate model currently, as it models the dynamics of a neuron's ion channels through three differential equations, each representing one ion channel. However, it is too complex to implement in an SNN. Therefore efforts were done to approximate this model through simplification. Examples are models like the Izhekevich neuron [20] that reduce the Hodgkin-Huxley model to two dimensions, and integrate-and-fire neurons [21]. SNNs usually employ leaky integrate-and-fire [17] or spike response neurons (a generalised form of the integrate-and-fire model) [22], because they are efficient in computation

[3]Image source: https://en.wikipedia.org/wiki/Refractory_period_(physiology), last accessed (17/11/2021).

and rather simple to model. This work employs the leaky integrate-and-fire neuron model which is described in the following.

Leaky Integrate-And-Fire

Leaky integrate-and-fire (LIF) neurons are the simplest of the integrate-and-fire neuron models [11, 17]. Integrate-and-fire neurons model biological neurons with two mechanisms.

Firstly, the *Integrate* mechanism dictates the computation of a neuron's membrane potential evolution over time. This is defined through a differential equation. In the case of LIF neurons, the membrane potential u is given by this linear differential equation:

$$\tau_m \frac{du}{dt} = -[u(t) - u_{\text{rest}}] + RI(t) \tag{2.1}$$

where $u(t)$ gives the membrane potential at time t, u_{rest} defines the resting potential of the membrane, $RI(t)$ describes the amount by which the membrane potential changes to external input (R being the input resistance and $I(t)$ the input current), and τ_m is the time constant of the neuron.

Secondly, the *Fire* mechanism controls the spike generation of the neuron. LIF neurons fire when the membrane potential u crosses a defined threshold θ from below. The firing time $t^{(f)}$ is given by:

$$t^{(f)} = \{t | u(t) = \theta \wedge \frac{du}{dt} > 0\} \tag{2.2}$$

After firing, u is reset to the reset potential u_r, which is smaller than u_{rest}. This mechanism reflects relative refractoriness, as it lowers the chance of the neuron firing again immediately. Without input, the membrane potential recovers to u_{rest} after a certain time, as given by (2.1).

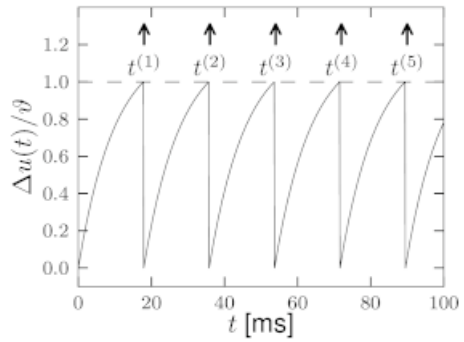


Figure 2.3: Spikes $t^{(f)}$ as generated by a LIF neuron for a constant input. Threshold θ is denoted in the dashed line (from [17]).

LIF neurons are simple to compute and implement but do not account for absolute refractoriness, i.e. the period in which neurons are not able to fire directly after a spike. Therefore, given a sufficiently strong input, LIF neurons can fire consecutively. Due to their simplicity and efficient computation, they are commonly used in SNNs. However, LIF neurons also oversimplify the biological processes.

2.1.3 Neural Code

SNNs use specific neural codes. Unlike artificial neurons, spiking neurons receive and produce spike trains, or binary sequences, as in- and output. As data is often real-valued, it is converted to a suitable format through so-called neural coding schemes in SNNs. Mainly three different neural codes are distinguished: Rate coding, temporal coding, and population coding. Based on the neural code, data is presented in different spike patterns. Additionally, the neural code influences the complexity of the problem [23].

Whereas rate coding assumes the information about a stimulus to be coded in the number of times a neuron fires in a defined time window, temporal and population coding consider the exact spike timings to also carry information. Therefore, the latter two are closer to biological reality. In population coding, a stimulus is translated into spike times using a group of encoding neurons. This group is called a population. Therefore, the information about the stimulus is encoded by multiple neurons and the SNN requires an additional encoding layer [24]. Temporal coding translates the input directly to a certain spike time and is commonly used for time series data, which already exhibits a temporal dimension. Therefore, this work uses temporal coding for the SNN models as well. Furthermore, since the number of related works in explanation methods for SNNs is strongly limited, this work targets a rather simple method that shall be widely applicable. An additional population encoding layer would entail the additional efforts of inverse coding to relate the population to an input dimension, while temporal coding allows for direct mapping. Therefore, the choice of temporal coding prevents specific efforts concerning the neural code in the targeted explanation method.

Temporal Coding

Temporal coding assumes the information about a stimulus to be encoded in the specific firing times of a neuron [17]. A simple temporal code is latency coding (Figure 2.4), where the information about the stimulus is encoded in the time between stimulus presentation and the first produced spike firing time [17, 23]. This coding scheme is also often referred to as *time to first spike* (TTFS). It is based on the idea that the spiking pattern of a neuron changes when the stimulus changes, e.g. when a human's gaze jumps during reading. Therefore, the information is in the latency to the first spike upon stimulus change, where a short latency is linked to the strong stimulation of a neuron. The following spikes within a time window are irrelevant. In neuron models, they are often suppressed by defining a long refractory period.

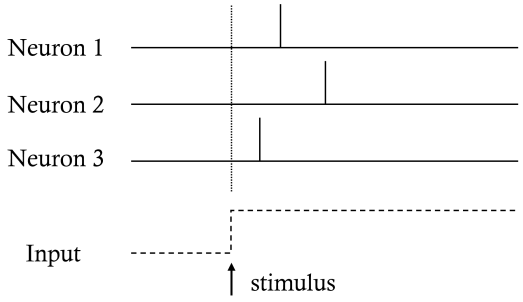


Figure 2.4: Latency coding of three neurons. The dashed line represents the stimulus, with a change at the step. The third neuron responds strongest to this change because it fires first [17].

2.1.4 Learning algorithm

No prominent learning method currently exists for SNNs and the majority of SNN research is directed towards finding an efficient learning algorithm. The difficulty in transferring learning algorithms from ANNs to SNNs lies in the non-differentiable nature of spiking neurons, caused by their spike and reset mechanisms. As a consequence, error backpropagation, which is the established learning algorithm in ANNs, is not applicable. Error backpropagation relies on error gradients which are computed from an error function using the chain rule of derivatives [25]. There are different approaches in the literature to overcome the non-differentiability of spikes and facilitate learning, ranging from unsupervised methods [26] to more complex evolutionary algorithms, reinforcement learning, and Hebbian learning [19]. Furthermore, research also looks at converting a trained ANN to an SNN so that error backpropagation can be used [27], smoothed networks or surrogate gradients [24, 28, 29]. This work uses surrogate gradient learning.

Surrogate Gradient Learning

Surrogate gradient learning overcomes the non-differentiability of spiking neurons by substituting the undefined gradient by a surrogate in the backward pass through the network [29]. The surrogate gradient acts as a continuous relaxation of the true gradients, without changing the model definition. This allows optimisation of the network with error backpropagation using gradient descent, thus enabling the training of multi-layer networks. Several possible choices for surrogate gradients exist (Figure 2.5) and were applied in several studies with SNNs using surrogate gradient learning.

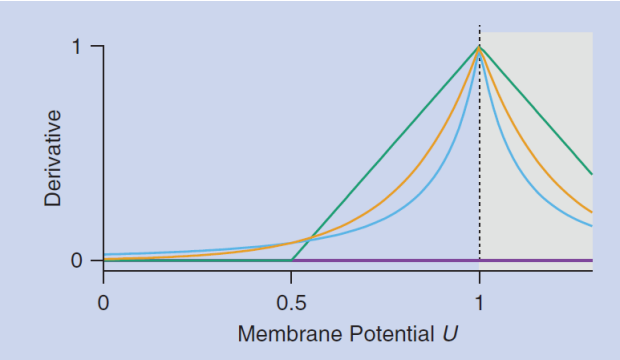


Figure 2.5: Different surrogate gradients [29, p. 56], rescaled to $[0, 1]$ (Stepwise function in violet, piecewise linear in green, exponential in yellow, fast sigmoid in blue).

Zenke and Vogels [30] studied the robustness of SNNs trained with surrogate gradients with regards to the shape and scale of the surrogate function. They found that the shape of the gradient, i.e., the choice of the surrogate derivative does not have a large effect on learning. However, the scale of the surrogate function should not be too large to prevent exploding or vanishing gradients during training.

2.1.5 Building Spiking Neural Networks for Research

As the computations of SNNs depend on their temporal dynamics which are often characterised through ordinary differential equations, specific SNN simulators are usually required to build SNN models. Already in the programming language Python, several different simulation environments in the form of libraries exist (e.g., *Brian2* [31] or *BindsNET* [32]). Usually, simulators

have different focuses, e.g. *Brian2* has strong applicability in neuroscience and *BindsNET* is more oriented toward machine learning applications. Unfortunately, *BindsNET* does not implement surrogate gradient learning at the time of this work and learning using out of the box local learning methods did not yield promising results in preliminary experiments. Therefore, neither simulator is used. However, SNNs can also be interpreted as recurrent networks in discrete time. This enables the implementation and SNN training using libraries and toolboxes for ANNs. Therefore, this work implements SNN models as recurrent neural networks in discrete time using *PyTorch* [33] similar to the work of Neftci et al. (2019) [29].

SNNs as Recurrent Networks

SNNs with LIF neurons and current-based synapses can be formulated as recurrent networks with binary activation functions by considering the dynamics of the synaptic currents and membrane potential in discrete time [29].

The LIF neuron, as explained in section 2.1.2, is defined through a linear differential equation of the membrane potential in time $u(t)$, where $u(t)$ acts as the leaky integrator of the input current $I(t)$. Therefore, synaptic currents, i.e. the currents that flow through the synapses of connected neurons, follow specific temporal dynamics. Assuming that different currents follow a linear summation, a first-order approximation of the synaptic current dynamics yields an exponentially decaying current following input spikes $S_j^{(l-1)}$. In other words, the dynamics of synaptic currents decay exponentially in time, and are increased linearly by the synapse weight W_{ij} and recurrent weight V_{ij} at every input spike to the neuron:

$$\tau_{syn}\frac{dI}{dt} = -I(t) + \sum_j W_{ij}^{(l)} S_j^{(l-1)}(t) + \sum_j V_{ij}^{(l)} S_j^{(l)}(t) \tag{2.3}$$

To view these dynamics in discrete time, first, the output spike train $S_i^{(l)}[n]$ of the LIF neuron is formalised in discrete time, where n denotes the discrete time step:

$$S_i^{(l)}[n] = \Theta(u_i^{(l)}[n] - \theta) \tag{2.4}$$

Setting the firing threshold $\theta = 1$, the above equation describes the spike train using a Heaviside step function Θ, so that the values in $S_i^{(l)}$ evaluate to $\in \{0, 1\}$, so either spiking at n or not. Then, for a small time step > 0, a resting potential $u_{\text{rest}} = 0$, and an input resistance of $R = 1$, the synaptic current dynamics and membrane potential dynamics can be formulated in discrete time as follows:

$$I_i^{(l)}[n+1] = \alpha I_i^{(l)}[n] + \sum_j u_i^{(l)} S_j^{(l-1)}[n] + \sum_j V_i^{(l)} S_j^{(l)}[n] \tag{2.5}$$

$$u_i^{(l)}[n+1] = \beta u_i^{(l)}[n] + I_i^{(l)}[n] - S_i^{(l)}[n] \tag{2.6}$$

In the above equations, $\alpha = \exp(-\Delta t/\tau_{syn})$ and $\beta = \exp(-\Delta t/\tau_{mem})$ describe the strength of exponential decay of the synaptic current and membrane potential respectively. According to [29], equations 2.5 and 2.6 describe the dynamics of a recurrent network, where the membrane potential is the cell state that is calculated by considering the synaptic input currents.

2.2 Foundations of Explainable Artificial Intelligence

Explainable Artificial Intelligence is a large area of research that spans various methods addressing the wide topic of explaining models and predictions. In this section, an overview of the vocabulary definitions and taxonomies are given as prerequisite terminology to this thesis.

2.2.1 Terminology

Explainable Artificial Intelligence (XAI) concerns itself with explaining the decisions and predictions made by machine learning models to humans. XAI is an active field of research since around 2015, and many models, as well as methods, exist that provide explanations and interpretability on different levels [6]. The term *Interpretable Machine Learning* (IML) is also used to describe this field and will be used interchangeably in the frame of this work.

The *interpretability* of a machine learning model refers to its ability to be understood by a human. There is no mathematical definition of it, rather it is measured by the degree of human understanding [34]. Barredo et al. (2019) [35] define interpretability similarly, as a passive characteristic of a model, that is defined by how much sense a model's behaviour and decisions make to a human. *Explainability*, in contrast, is an active characteristic of a model which describes the behaviour of the model that actively contributes to its decisions being human-understandable. It also does not have a mathematical definition, and it is unclear how model explainability is measured. Instead, it is an attribute a model has or not, as it is an active characteristic. Nevertheless, both concepts are devoted to making machine learning models understandable to humans. Consequently, they are at the core of IML and XAI, which aim at providing a suite of explanation methods and models that are transparent and understandable for humans in their predictions and decisions [36, 35].

Miller (2019) [34] defines an *explanation* as an answer to a Why-question. In this sense, the main question answered by explanations of IML and XAI methods can be formulated as *Why does a model make a (certain) prediction?* A good answer to this question is a good explanation, but the requirements for such in literature are not very specific. Powerful explanations should be general [10, 36], meaning that their applicability holds for many examples. Another requirement for a good explanation is its clarity. It should leave little to no room for user interpretation, which could lead to misunderstandings caused by poor clarity of the explanation [35]. In accordance, the explanation should focus on the user audience [36]. The level of explanation varies with the background knowledge and prior beliefs of the audience, therefore explanations have a social aspect that should be considered. Thus, no set requirements for a good explanation exist, rather it depends highly on the data used, and the audience that the explanation targets.

2.2.2 Taxonomies

Due to the general nature of the definition of XAI, many methods and models fall under this topic. These can be divided according to a general taxonomy of three criteria [7] (Figure 2.6).

First, an explanation method is specified by the *scope* of its given explanation, which can be either local or global. In a local scope, an explanation is provided for an individual prediction of the model, whereas a global explanation gives insights into the global model behaviour. The latter is quite difficult to achieve, as it is about providing a global understanding of how input features and the model are related to an outcome distribution. Therefore, the complexity of the task increases with the number of features and parameters of the data and model used. Second, methods are distinguished by the *moment of method implementation*. On the one hand, methods can be intrinsic. This means that interpretability, or rather explainability, is already a

part of the model itself: It is intrinsically explainable. Often, intrinsic methods and models have restricted complexity (e.g. Decision Trees). On the other hand, methods can be applied post-hoc, meaning that they are used at model inference. Third, explanation methods are classified according to their *model specificity*, where a method is either model-specific or model-agnostic. Model-specific methods are limited to a specific type of model, and thus cannot be used for other types. Model-agnostic methods, however, are general methods that can be applied to any model. Usually, these methods are post-hoc. In addition to these, Molnar (2019) [36] also identifies the result of the interpretation method as a criterion for discriminating different methods. Summary statistics as an explanation are differentiated from visualisation methods, as well as certain data points for explanation (i.e. representative data points for a prediction) and intrinsically interpretable models.

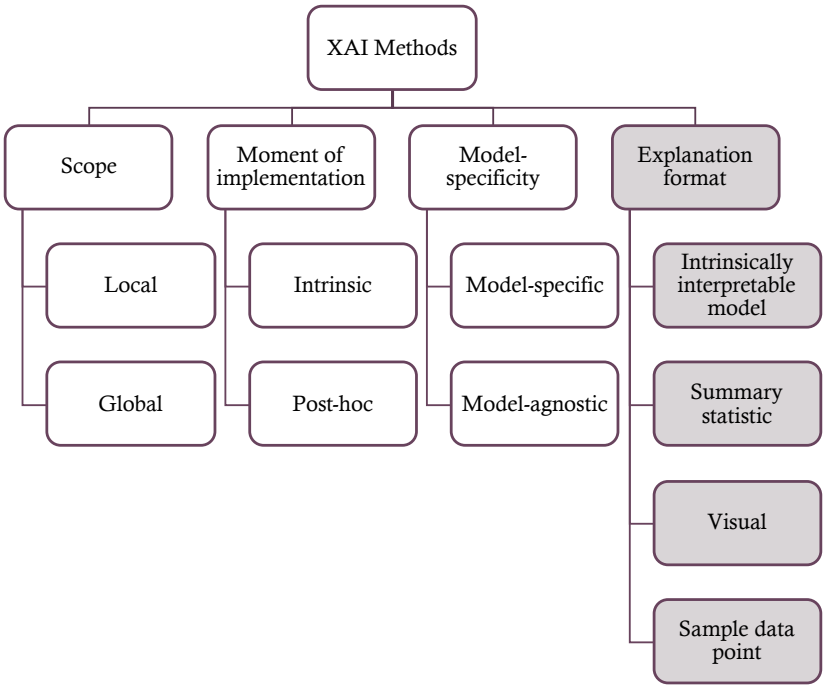


Figure 2.6: General XAI taxonomy [7] and additional criteria (shaded) by Molnar (2019) [36].

Additionally, Guidotti et al. (2019) [10] define four problems in XAI, according to which the suite of models and methods can be classified (Figure 2.7). The *model explanation problem* addresses the global explanation of a model. The emphasis on this problem is put on global interpretability. Therefore, methods that solve the model explanation problem provide an explanation that makes a model's decision logic understandable to humans. Guidotti et al. (2019) [10] mention solving this problem by finding a transparent model, which mimics the behaviour of the original black box model, and therefore can give global explanations. The *outcome explanation problem* is concerned with the explanation of a model's prediction for a certain input. Therefore, this problem is essentially addressing local explanations, as opposed to the first problem. The *model inspection problem* targets the understanding of internal model behaviour given a certain input. For example, an inspection of the learned parameters of a neural network gives insight into the internal model behaviour. So, this problem is also overlapping with the other problems. A method can therefore be categorised into multiple problems as well. The last problem is the *transparent box design problem*, which is about designing a transparent model, that is human-understandable on a local and global level by default.

The explanation method developed in this thesis generates local, post-hoc explanations that shall be model-agnostic to temporally coded SNN models and address the outcome explanation and model inspection problem. The provided explanation is a feature attribution explanation,

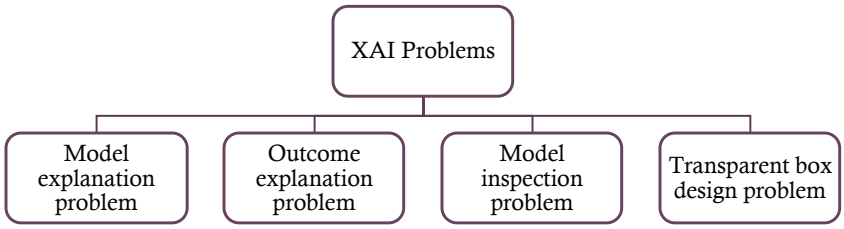


Figure 2.7: General XAI problems [10].

thus the explanation format is a two-dimensional heat map that is visualised for presentation to the user.

3 RELATED WORK

The thesis provides novel research on explanations from SNNs trained on a time series classification task. To give an overview of the related fields, this chapter first presents related methods. Additionally, literature linked to XAI with time series and SNN architectures is presented to understand common approaches to explanations and model development respectively. This enables the positioning of the thesis work in the fields of research regarding SNNs as well as XAI.

3.1 Interpretable Spiking Neural Networks

The number of previous studies concerning explanations for SNNs is currently quite limited. Very few works have studied this topic, and it is a rather unexplored area of research. This section highlights 2 methods: the first addresses global interpretability through finding feature strength functions, whereas the second provides a local explanation based on interspike intervals.

3.1.1 Global Interpretability through Feature Strength Functions

Jeyasothy et. al (2019) [37] presented one of the first interpretability methods for SNNs[4]. They identify interpretable knowledge for a specific SNN model based on SEFRON, the *Synaptic Efficacy Function-based leaky integrate-and-fire neuRON* [38].

SEFRON is a neuron model that can solve binary classification tasks with one LIF neuron and time-varying synaptic efficacies, meaning time-dependent weights of synapses (Figure 3.1). Therefore, the synapse values are determined by a continuous function over time. SEFRON synapses are inspired by an observation from the field of neuroscience: The possibility of an inhibitory synapse to switch to an excitatory synapse and vice versa[5]. The work uses population coding for neural coding. The input spikes are multiplied with the synaptic efficacy at time t to determine the postsynaptic potential (PSP) in the postsynaptic neuron. The first output spike is then used for classification, where the class is predicted based on the spike time of the output neuron. The model is trained using supervised spike time-dependent plasticity[6] (STDP) with target synapse strengths, that represent the ratio of the firing threshold to the ideal PSP for the correct classification.

The extraction of interpretable knowledge from a multi-class SEFRON model (MC-SEFRON) using different UCI machine learning datasets (e.g. Iris) and the handwritten digits MNIST dataset was demonstrated in [37]. MC-SEFRON differs from SEFRON in terms of the output layer size,

[4] This paper is published as a preprint.

[5] This switching has particularly been observed in developing brains and is referred to as the gamma-aminobutyric acid-switch [38].

[6] Learning rule that can be seen as a spike-based form of Hebbian learning. Synapses are strengthened if the time interval between the output spike and input spike is short, and weakened otherwise [39].

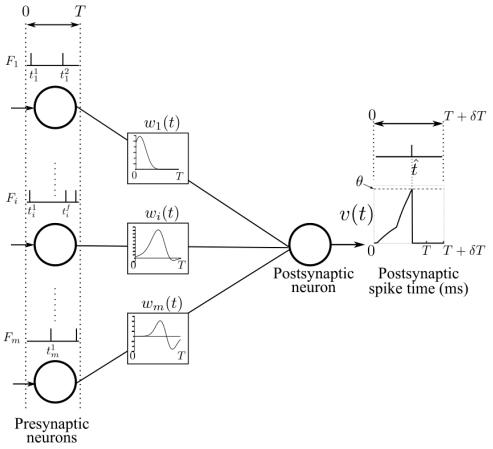


Figure 3.1: Single SEFRON model with time-varying synaptic weights $w_i(t)$ [38, p. 1233].

where the model has as many output neurons as classes (Figure 3.2). Consequently, the earliest spiking output neuron determines the predicted class $\hat{y}$ (3.1), where θ_j is the threshold of the j-th neuron and $U_j(t)$ is its membrane potential.

$$\hat{y} = \arg\min_j \min\{t | U_j(t) \geq \theta_j\} = \arg\min_j \min\{t | \frac{1}{\theta_j} U_j(t) \geq 1\} \tag{3.1}$$

In all other aspects, the computations of the MC-SEFRON model are derived from the SEFRON model. Thus, the network is shallow, uses time-varying synaptic weights determined by a weight function over time, and the learning is also based on supervised STDP with target synapse strengths. Furthermore, the input is encoded using a population coding scheme.

To extract interpretable knowledge from the MC-SEFRON SNN, this population coding scheme is made use of. Population coding can be viewed as a function $G(x)$ of an input x, which results in a spike train s, according to the defined size of the population and receptive field of each population neuron. By using the inverse of G, it is possible to map spike trains back to the input feature domain due to the unique solution of this problem:

$$G^{-1}(s_i) = \{x_i | G(x_i) = s_i\} \longrightarrow s_i = G(G^{-1}(s_i)) = G(x_i) \tag{3.2}$$

Therefore, Jeyasothy et al. (2019) [37] define so-called feature strength functions (FSF) $\psi_i(x_i, j)$ of an input feature x_i and output neuron j by replacing the spike time s_i^r of the r-th population neuron of x_i with $G(x_i)^r$ in the computation of the membrane potential. The FSF reflects the relation between input and output that is learned by the MC-SEFRON SNN.

Hence, the FSF is a function of the input, which is in a human-understandable domain, instead of the temporal domain of spike trains. It shows the relationship between an input feature and the output classes, thus providing global model insights and addressing the model explanation and model inspection problem [10]. Moreover, the FSFs can be used in a classification task as they are specified for each connection between the input and output neurons of the network they are derived from. The classification then occurs according to the strongest aggregated feature strength between a given input x of class k and the output classes:

$$\hat{y}^* = \arg\max_j \sum_{i=1}^{m} \psi_i(x_i^k, j) \tag{3.3}$$

As FSFs can be utilised for the same classification task (Figure 3.3), their reliability of the inter-

pretable knowledge is validated using this property. Experiments with multivariate (several UCI machine learning datasets), image (MNIST), and time series data (EEG) show a minimal loss in prediction performance, thus validating the reliability of explanations provided through FSFs.

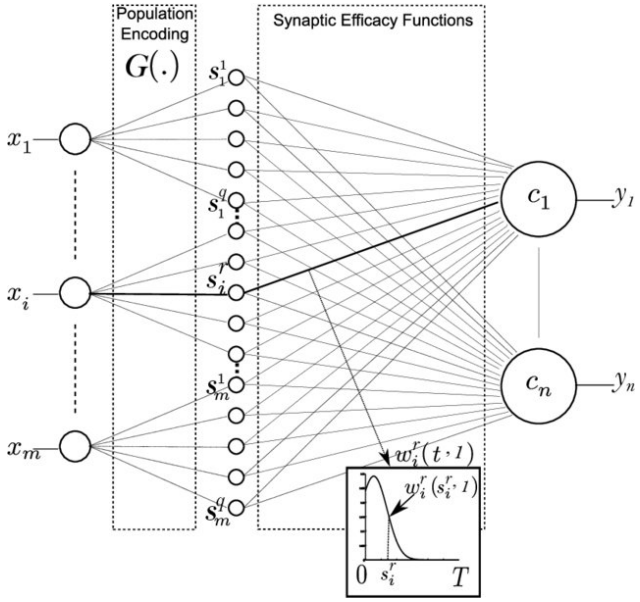


Figure 3.2: MC SEFRON model with population coding of input x_i [37, p. 4].

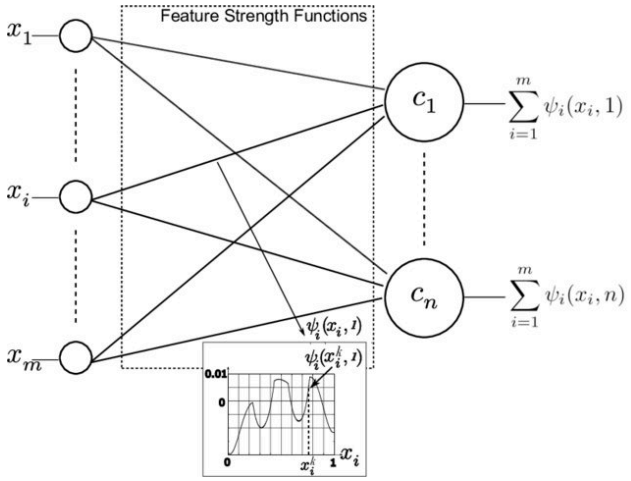


Figure 3.3: Classification model with FSFs extracted from MC SEFRON model from inverse population coding [37, p. 8].

In conclusion, the FSFs extracted from the MC-SEFRON model are a global explanation method, which is model-specific to SNN models with time-varying synapses and population coding and addresses the model inspection problem. It is the first work that highlights the requirement for SNNs to be explainable and showed how the spike domain and input domain can be bridged by an inverse mapping of the neural code. The approach taken in this thesis differs greatly from the FSF explanations [37] as it highlights a different side of XAI for SNNs. Instead of a global explanation for a specific SNN architecture, a local explanation is targeted, which explains a certain decision taken by the model. Moreover, the synapses are fixed over time so that the proposed method in this thesis applies to a wider range of SNN models. The proposed method is agnostic to all temporally coded SNN models, regardless of their architecture, while FSFs apply to shallow SNNs with one computational layer and require time-varying weights. Furthermore, this thesis also addresses the quality of the proposed explanation method by pro-

viding a thorough evaluation of an explanation that includes aspects beyond the reliability of an explanation.

3.1.2 Local Explanations with the Spike Activation Map

A recent work by Kim and Panda (2021) [40] presents a local explanation method for SNNs called *Spike Activation Map* (SAM). SAM is a visual heatmap explanation with a temporal dimension and was applied to deep convolutional SNNs with LIF neurons. SAM makes use of the biological observation that short time intervals between the spikes of a neuron likely carry information as they have a high chance of causing a postsynaptic spike. Based on this observation, the so-called *Temporal Spike Contribution Score* (TSCS) (3.4) is defined. The TSCS describes the contribution of a previous spike time $t^{(f)}$ to the current time t in one neuron, formulated with an exponential kernel with the steepness parameter γ.

$$T(t, t^{(f)}) = \exp\left(-\gamma|t - t^{(f)}|\right) \tag{3.4}$$

As the TSCS is formulated for one single spike time $t^{(f)}$, an additional score is computed to achieve the explanation. The *Neuronal Contribution Score* (NCS) (3.5) sums all TSCS of the previous spikes of one neuron at time t. Thus, it quantifies the contribution of a neuron to its downstream neuron's spiking behaviour (Figure 3.4). A high NCS means that many spikes are fired within a short time window, while a low NCS indicates a small number of spikes distributed in time. Let P be the set of previous spike times of a neuron. That neuron's NCS at current time t is then:

$$N(t) = \sum_{t^{(f)} \in P} T(t^{(f)}, t) \tag{3.5}$$

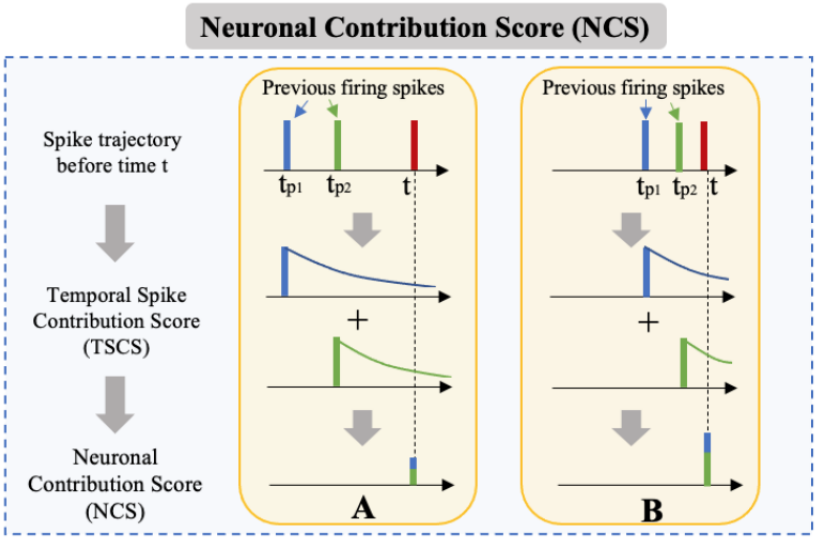


Figure 3.4: Visualisation of the NCS computation [40, p. 5].

Using the NCS of all neurons of the network, the SAM is computed at current time t by a forward pass in the network through multiplication with the NCS at t. The visualisation is determined through a sum of the NCS' across the channel axis of a convolutional layer. Let $S(t)$ denote the input spike pattern of the data up until t. Kim and Panda (2021) [40] then define the map M_t

to be a sum of the product of input and NCS over all channels k, as they demonstrated their method on coloured image data.

$$M(t) = \sum_k N^{(k)}(t) S^{(k)}(t) \tag{3.6}$$

Thus, at each time step t, a different SAM is generated, highlighting the parts of the input which contribute more to the prediction. It is noteworthy that this explanation can be computed without a target label, as it is not gradient-based like other heatmap explanation methods for ANNs.

In their work, explanations were generated for rate-coded image data (Tiny-ImageNet) in convolutional SNNs trained with surrogate gradient learning as well as ANN-SNN conversion and compared against another heatmap method for ANNs, namely Grad-CAM. It was found that SAM provides higher variance in the heatmap as it does not suffer from using an approximated gradient, which smoothes the heatmap. Additionally, SAMs extracted from the SNN trained with surrogate gradient learning were more accurate (i.e., more similar to heatmap generated with Grad-CAM of ANN) than from the SNN which was converted from an ANN.

In conclusion, SAM provides a local explanation method for SNNs which does not require back-propagation, i.e., does not require gradients. The effectiveness of this method was shown in [40] for rate coded image data in convolutional SNNs. Interestingly, the components of this method (i.e., NCS and TSCS) are model-agnostic to SNNs because they are based on the spike patterns of spiking neurons. This thesis targets a similar explanation method for temporally coded SNNs and employs Kim and Panda (2021)'s [40] TSCS and NCS. In this thesis research, the definition of the NCS is extended by additionally considering the learned weights of an SNN directly in the calculation. Furthermore, we look at temporally coded time series data as opposed to rate-coded image data, thus providing a novel contribution to local explanations using spike patterns in SNNs. Additionally, this work also focuses on the qualities of an explanation on time series data. This thesis proposes a set of evaluation metrics for the generated explanation and investigates the quality of the explanation.

3.2 Explainable AI with Time Series

In contrast to the sparsity in related work with regards to explanations from SNNs, XAI is an active area of research for ANNs, with increasing research focus in recent years. Overall, works in the field of XAI for models trained on a time series classification task are less common than for other data types, e.g. image or text data. A large factor for this is the non-intuitive interpretation of time series. Compared to an image, for example, it is less straightforward for a human to identify relevant parts of the data with time series [41]. However, explanations for time series data remain relevant as this type of data is prevalent in many sensitive application areas, e.g. health, traffic, or natural disasters [42].

3.2.1 Explanation Methods with Time Series

There are several different approaches to explanations with time series data, which have different scopes, model-specificities, and explanation formats. Furthermore, different types of classifiers and time series datasets can be found in related literature, which impact the explanation method used. This section presents a short overview of related work of local explanations.

One type of local explanation for time series can be found in literature in the form of sample data point explanations. One example is the work of Ates et al. (2020) [43], who generated counterfactual explanations for a high-performance computing telemetry dataset. They targeted

faithful, robust, and human-comprehensible explanations. By using a counterfactual example that is similar to the original input and presenting it in comparison, the authors want to overcome the complexity and non-intuitiveness of time series data. Another example is Kuesters et al. (2020) [44], who criticised heatmapping and visual explanations for time series data. Inherent properties of time series, e.g., trends or seasonality, are often not able to be localised in an input. Instead, these properties are often spread over the input. As they can be important to explain a prediction, the inability to localise trends, for example, is a drawback. The authors propose the use of conceptual explanations, which are mask-based approaches to explanations. However, instead of masking input regions, they mask input properties using different filters, thus aiming at explanations that highlight these concepts. These aforementioned explanations mention relevant downsides to visual explanations for time series data regardless of the used model. They show that by using sample-based explanations, these limitations can be overcome. In the frame of this thesis, SNNs are used which can inherently process the temporal domain of time series data. As this work is one of the firsts of its kind, not much is known yet about the effect of e.g. seasonality in explanations extracted from SNNs. Therefore, this thesis aims at solving a more fundamental problem first, therefore aiming at feature-based explanations.

Many local explanation methods for time series tasks provide visual explanations, such as [45] and [46]. These works provide a heatmap explanation, in which the importance of input features and the time step are indicated with varying intensities. Assaf et al. (2019) [46] present a backpropagation-based explanation architecture using a similar approach like Grad-CAM[7] on the axes of time and input feature (Figure 3.5). They visualise the attention of their model in the granularity of time step per input feature for a multivariate time series classification task, resulting in an interpretable heatmap per input.

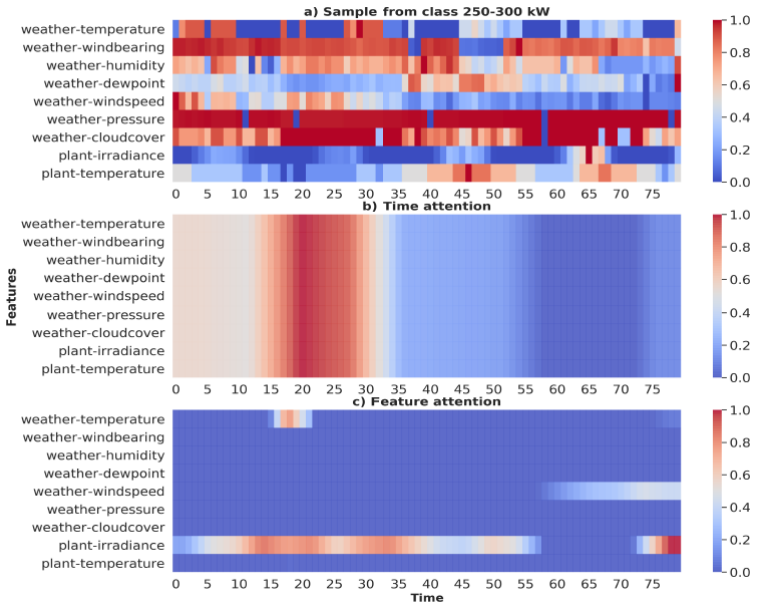


Figure 3.5: Example data and explanation from Assaf et. al (2019) [46, p. 6489] for an energy forecasting task for photovoltaic power plants using observations of the plant and weather. a) displays the data sample. The attention explanation across the axes of time and input features is visualised in b) and c) respectively. b) represents the joint contribution of the input features across time whereas c) gives more insight into the contribution of the single features.

Kono, Yamaguchi and Nagao (2020) [45] take a different approach by utilising generative contribution mapping, a method from the field of explanations for image data. Generative contribution

[7] Gradient-weighted Class Activation Mapping. A common heatmap explanation method for images [47].

mapping creates importance maps per class, which are inherently interpretable as their weights can be treated as partial regression coefficients. The authors applied this method on time series data as a 1D image and obtained an importance map showing the class probabilities as an explanation through the aggregation of the different importance maps.

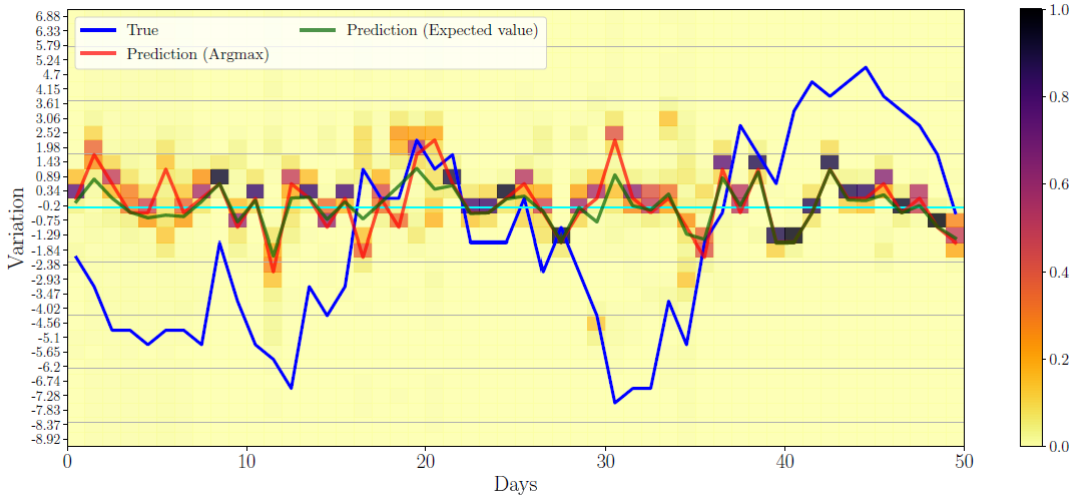


Figure 3.6: Example explanation from Kono, Yamaguchi and Nagao (2020) [45, p. 4100] for a time series forecasting task (prediction of crude oil prices using price variations in a day as data). The blue line corresponds to the ground truth, the red line and green line correspond to the prediction using either argmax or expectation. The explainability is given through the square highlights in the background. These regions inform the user of the class probabilities, where a darker colour represents a higher probability for a class.

The explanation method from this thesis also aims at providing a visual explanation in form of a heatmap that highlights the attribution of the input at a certain time. Following the related works, the attribution in this work will also be retrieved on a granularity of time and input dimension. As the underlying model is different, and this thesis aims to provide an SNN-specific explanation method, only concepts can be transferred and a new explanation method must be formulated.

3.2.2 Desired Properties of Explanations for Time Series

Unlike the predictive performance of machine learning models, a model's interpretability does not have clear performance metrics. There is also no consensus about a set of metrics and requirements that explanation methods need to fulfil. Instead, the requirements of interpretability are dependent on the task, the model, and the target audience [6]. This lack of specificity in requirements for explanations also applies to explanations for time series.

Fauvel, Masson and Fromont (2020) [42] provided a framework to particularly assess the performance of a model on multivariate time series (MTS) data, specifically including the model's explainability. They argue that machine learning models overall should be evaluated with regard to their explainability in addition to the predictive accuracy. However, their framework is quite high-level, mainly categorising models into the taxonomy of XAI [7]. Detailed metrics for faithfulness (i.e. does the explanation truthfully reflect model behaviour) are not given, instead, it is treated as a binary property that is false if a surrogate model is used for the explanation so that the framework applies to a large class of models. This demonstrates the heterogeneity of the evaluations of explanations, also for models trained on MTS data.

A recent survey [48][8] defines the purpose of explanations in the provision and/or increase of user

trust in a black box model for time series. Several building blocks act as a foundation for trust: Stability (i.e., model is resistant against small, naturally occurring perturbations), robustness (i.e., adversarial robustness), and confidence (i.e., explanation highlights prediction confidence of the model). Therefore, this details the desired properties of explanations for time series in a slightly higher level of detail than the previously mentioned work [42].

Regardless of what task explanations are generated for, there are more aspects than the aforementioned to be considered [36]: An important part of the quality of explanation lies in its comprehensibility. To be comprehensible, an explanation has to be human-understandable with regard to the target user group.

These depicted properties (i.e., faithfulness [42], stability, robustness, confidence [48], and comprehensibility [36]) of explanations are taken as a basis for the evaluation of the proposed method of this thesis. As they are not directly and straightforwardly related to specific metrics, this is the topic of the second sub-research question.

3.3 Common Architectures of Spiking Neural Networks

Even though the idea of SNNs exists for roughly 25 years already [11], there is no consensus in the literature about an ideal architecture for SNNs, including the choice of neuron model and neural coding. Especially efficient learning methods for SNNs are an active field of research. However, to identify an appropriate and representative architecture for the underlying models for this research, related works using SNNs for classification tasks in current SNN research have been identified. Several scientific search engines and libraries were queried on 23/02/2021 with the phrase *spiking neural network classification*, namely *Google Scholar, Scopus, arXiv, ACM Digital Library, IEEE Xplore, ScienceDirect*. These libraries were selected as they appeared to carry most publications around SNNs from prior literature research. The search results were sorted by relevance, and of each search engine, the top five publications in English that studied an SNN classifier were accessed to get an impression of common classification tasks for SNNs. Three main groups of tasks can be identified from these retrieved papers, including time series classification. A summary overview is shown in Table 3.1 and the full overview can be found in appendix A.

	Tabular	**Image**	**Time series**
Number of papers	7	9	18
Publication years	2001 - 2021	2015 - 2020	2010 - 2021
Most common neuron model	IF variations	LIF	LIF
Most common neural code	Rate	Rate	Temporal
Network depths	1-2 Layers	1-6 Layers	1-6 Layers
Learning methods	Various	Various	Various

Table 3.1: Summary overview of surveyed papers.

The first group of tasks study SNN performance in tabular data classification, mostly with common UCI machine learning datasets, such as the Iris dataset [49]. These datasets have likely been used as a benchmark for SNNs, as they are standard datasets for testing classification methods. Thus, SNNs become comparable with other machine learning algorithms and ANNs for these tasks. It is noteworthy that mainly integrate-and-fire neurons have been used, and no temporal coding is used. There is a tendency toward shallow architectures with at most one hidden layer. Little layers seem to suffice for solving tabular data classification tasks with SNNs.

[8] This paper is currently under review.

Additionally, it can be observed that different learning methods are used, beyond supervised learning (e.g., evolutionary algorithms). In the second group of tasks, the classification of image data is investigated. Several works have been found to test SNN classification performance for common benchmark datasets such as MNIST and CIFAR. Also, these tasks were used to explore an SNN's power and to compare it, especially to convolutional ANNs, which are known to deal well with image data. Through these studies, a convolutional SNN architecture similar to the ANN counterpart has emerged. In image classification with SNNs, rate coding dominates the neural codes and LIF neurons are predominantly used. Similar to the works on tabular data, often shallow architectures but also more complex architectures like deeper convolutional SNNs are used. These SNNs are often inspired by the LeNet architecture from ANNs. The learning methods for image classification are various, even though a tendency towards STDP-based methods is noticeable. While tabular and image data are interesting to study for comparison to other models due to the existing benchmark datasets, the temporal dimension of data, which is particularly interesting for SNNs, is neglected. This is reflected in the prevalent use of rate coding in the related works on these tasks. The information about the data is often assumed to be encoded in the firing rate solely. As the targeted explanation method shall make use of the exact firing times, similar to [40], these types of data are not the first choice. Hence, this thesis research does not look into tabular or image data.

The datasets at the centre of the third group of common tasks found for SNNs are of time series nature, which is of interest to this thesis. Mainly signal data, such as EEG or audio signals are studied. However, also efforts to create a spatio-temporal domain version of MNIST (N-MNIST) has been realised and is subject to SNN research. This list is larger than the other two groups, indicating that SNNs are especially of interest with time-domain data, which makes sense as SNNs can process spatio-temporal data. This finding supports the choice of a time series classification task in the frame of this thesis with regard to application relevancy. Nevertheless, it remains unclear in most papers whether the time series datasets have been processed as such (i.e., with temporal dependencies) or as tabular data (i.e., every timestep as an independent data point). Experiments were made with different neuron model types, with a dominance of integrate-and-fire neurons. All three main neural codes were utilised with a slight domination of temporal coding, and the variety of architecture is similar to the previous groups, i.e., rather shallow architectures. Especially in this group of surveyed papers, diverse learning methods are observed. Therefore, this thesis research takes the liberty of choosing a learning algorithm that achieves acceptable performance for the selected dataset.

4 USE CASE MODEL AND DATA

As the objective of this thesis is the extraction of local, feature-based explanations from SNNs, an essential prerequisite are the SNN models. In this chapter, the SNN models and dataset used in the frame of this work are presented.

4.1 Data

This section details the dataset used in the frame of this thesis, as well as the preprocessing steps that have been performed with the objective of reproducibility of the work.

4.1.1 Dataset Description

The dataset used to train the SNNs is the Activities of Daily Living Recognition using Binary Sensors dataset [50] (ADL), which is openly available in the UCI repository. The ADL dataset is a multivariate time series dataset, which can be used for supervised or unsupervised learning. It consists of data that has been collected from a wireless binary sensor network installed in the homes of two subjects A and B. This dataset was chosen mainly because the task of activity prediction from these binary sensors does not require a domain expert to understand the data. The sensors in this dataset are expressive and easily human-understandable, e.g. if the *Bed* sensor is activated, the user is lying in their bed. Therefore, feature explanations on this dataset are assumed to be human-understandable.

The data was collected continually for 35 days and has a time granularity of 1 second, therefore there are no missing values in terms of sensor data. The dataset was labelled manually with one of ten labels describing the activity (e.g. Sleeping). As both subjects live in different houses, there are sensors in one house that do not exist in the other. However, the majority of sensors overlap because they are quite generic (e.g. bed or toilet), and thus the person-dependency of the collected data is limited and is assumed to be negligible.

In the use case of this thesis, the models shall learn a task of continuous activity prediction. This means that the network learns to predict a subject's activity at each time step of the data, which is each second in the case of the ADL dataset. To enable this type of task, the data labels specified in the dataset are broken down to labels per second. For example, if the dataset shows that the subject was eating *breakfast* on the first day at eight in the morning for 15 minutes, every second of these 15 minutes is also labelled *Breakfast* (Figure 4.1).

The dataset exhibits class imbalance. Figure 4.2 shows the class distribution across the activities in the dataset. There are activities such as *Spare_Time/TV, Sleeping, Leaving*, which occur significantly more often than others. This is noteworthy, as class imbalance potentially influences model training and performance. The class imbalance is enhanced by breaking down the labels to a time step granularity of one second.

For the experiments in the frame of this thesis, the dataset was split into a training (60%), val-

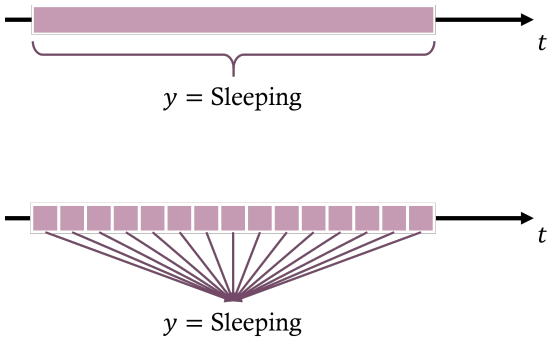


Figure 4.1: Visualisation of the label breakdown. If a time segment is labelled with y (upper part of Figure), e.g. *Sleeping*, each second of that time segment also has the label y on its own (bottom part of Figure).

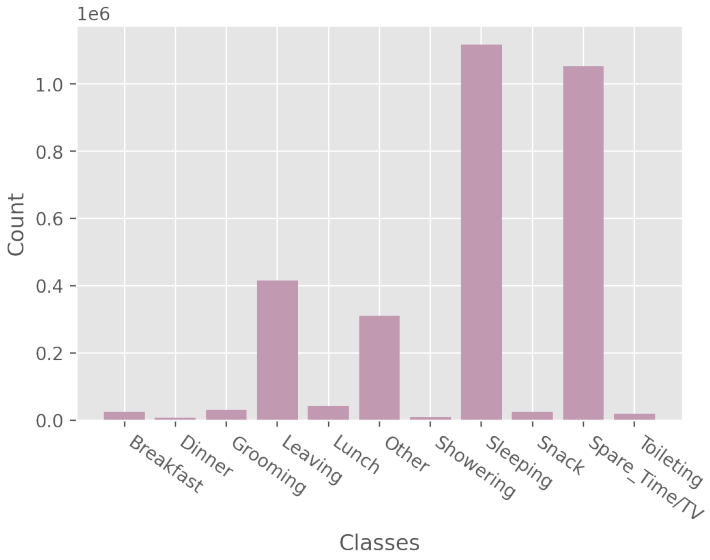


Figure 4.2: Class distribution of ADL dataset when broken down to a time granularity of one second after preprocessing. *Sleeping* and *Spare_Time/TV* are clearly the majority classes due to the relatively long nature of these activities. *Leaving* is also rather large because the the whole absence of the subject is considered as *Leaving*. Since the activities do not transition seamlessly into each other, the *Other* class is present between almost each activity. Consequently, the number of timesteps in *Other* is not small.

idation (20%), and test set (20%). Due to the non-i.i.d. nature of time series data, the split was performed sequentially to preserve the temporal dependencies of the data. For this, the dataset was first split into the subsets per subject, i.e. the first 60% of A's recordings are part of the training set as well as the first 60% of B's recorded data. Then, the subsets per subject are concatenated to build the final subsets (e.g. training sets of subjects A and B are concatenated to build the overall training set). Due to the long duration of the overall time series, this approach ensures that all parts of the day are present in each subset. Moreover, data from both subjects is represented in all subsets. Therefore, inherent properties like trends and periodicity are assumed to be present in all subsets.

4.1.2 Data Preprocessing

SNNs can inherently process time series data. Therefore, no heavy preprocessing of the ADL dataset is required before training the network. Instead, the binary sensor data is converted into a binary spiking input, which the model is trained on. Thus, the neural coding is a direct mapping of spike times.

Even though the dataset claims to be completely labelled [50], short time gaps were found which do not exhibit a label, even though sensor data was recorded. These cases were labelled as *Other* in this work, to avoid making assumptions about their true activity. *Other* is a class that is not expected to be learned by the model. Moreover, two cases[9] were found in which the specified end of the activity precedes the start of the same activity. These activities were excluded from the dataset as either the data collection or labelling is faulty. Since these are only two activities, their exclusion is assumed not to have a large impact, so that the valid entries are treated as one sequence with no gap.

4.2 Models

As mentioned in chapter 2, SNNs require a number of architectural design choices, similar to ANNs. In this section, the architectures and models used in this work are presented.

4.2.1 SNN Architecture Choices

The focus of this thesis lies in the extraction of explanations from SNNs. Therefore, rather simple architectures are chosen and built in the frame of this thesis:

Specification	**Choice(s)**
Neuron model	LIF
Neural coding	Temporal coding
Learning Algorithm	Surrogate Gradient Learning with Fast Sigmoid

Table 4.1: SNN architecture choices for this thesis.

For the neuron model, the underlying neuron model LIF is chosen for three reasons: First and foremost, the implementation is simple and fast. Secondly, related literature in SNNs has shown that simple neural models such as LIF are often used in SNNs. Therefore, this choice is valid when developing an SNN. Thirdly, the explanation method shall be model-agnostic with regards to SNNs, so that the specific SNN architecture is less important.

As SNNs can process temporal information directly, it makes sense to make use of this property to study explanations from SNNs. Therefore, temporal coding is chosen, as it assumes information about an input to be represented in spatio-temporal patterns, while rate coding neglects the exact firing times.

The SNNs use surrogate gradient learning from Neftci et al. (2019) [29] with a fast sigmoid surrogate gradient. This learning algorithm has proven effective for the ADL dataset in preliminary experiments, and with deep architectures with more than one computational layer.

[9] Entries with index 78 and 80 of label file of subject A.

4.2.2 Model Development

In total, three SNNs are built with differing depths: The first network has no hidden layers, thus only the output layer is a computational layer. It shall be referred to as *OneLayerSNN*. The second network has one hidden layer (*TwoLayerSNN*), and the third network has two hidden layers (*ThreeLayerSNN*). Three different models shall give insights into how explanations change upon the addition of hidden layers. The models were developed as recurrent networks with binary activations, using discretised formulas of the network dynamics[10], in accordance with Neftci et al. (2019) [29].

For all models, the input layer has the same size as the input dimensions, i.e. the number of sensors in the dataset, plus a bias neuron with a constant firing input (14 in total). The output layer is sized to the number of output classes (11), where each neuron corresponds to one class. The number of neurons in the hidden layers of the deep models are determined through hyperparameter tuning. All models have a fully connected feedforward architecture, where the membrane potential state $u(t)$ of a neuron is retained throughout the timesteps of a data sample by a recurrent definition. Unlike Zenke et al. (2018) [28] and Neftci et al. (2019) [29], this thesis deals with time series data that possesses temporal dependencies between data samples, i.e., activities in the case of the ADL dataset, that do not fulfil the i.i.d. assumption between data samples. The data is sequential, and past data samples can be relevant to the current data sample. Therefore, the model definition has been extended with regards to retaining the network variables (i.e., membrane potential $u(t)$ and synaptic current $I(t)$) in between any simulation runs. This means that the state variables of the model are initialised with the last states of the last simulation. This extension of the model definition presents a novel aspect to the SNN architecture of [29].

During model development, the data is presented to the model in a fixed number of simulation steps similarly to Neftci et. al (2019) [29]. Hence, each simulation run has the same duration and different time series lengths are not supported. This is done for reasons of computational efficiency and optimisation of memory usage during the run of the model. To represent the spiking data in the fixed duration format, the dataset which can be seen as one long time series is cut into non-overlapping sample time series of the same duration of 15 minutes (i.e., 900 seconds) (top part of Figure 4.3). The duration of the data is the same as the number of steps defined in a single simulation run so that each second of the dataset corresponds to one simulation step. By keeping the sample duration fixed, the resolution of the data during the simulation run is the same for each sample. Preliminary experiments showed that sequential training (i.e., by running the data of the first day up until the last day sequentially) is very slow and therefore not feasible. To take advantage of parallel processing options during training, the model is trained in batches. One batch runs a number of samples with the fixed duration of $T = 900$ in parallel (e.g., samples 1, 4, 7 are run in parallel in the example in Figure 4.3). After one batch, the model parameters are updated using gradient descent. It is assumed that the SNN models can learn the temporal dependencies without being presented with the whole time series (i.e. first to last day) in sequence. Nevertheless, an effort to ensure as much sequential order of the data as possible is made in the training process by the sample selection for a batch. The batch samples are selected to ensure the temporal order of the data (i.e., sample 1 is in batch one, sample 2 is in batch 2, and so on). Due to this, the data is possibly not fitting into a batch, so that it is padded with non-spiking data of the *Other* class. Furthermore, the synaptic currents and membrane potentials of the model are also retained in between epochs to keep the temporal dependencies between the epochs. To ensure the correct order in time, the dataset is rotated at each epoch during training (Figure 4.3). The gradients of the synaptic current and membrane potentials, however, are freed between batches during optimisation, to avoid an ever-growing

[10] Code available at: https://github.com/ElisaNguyen/tsa-explanations.

computational graph and with it, memory issues.

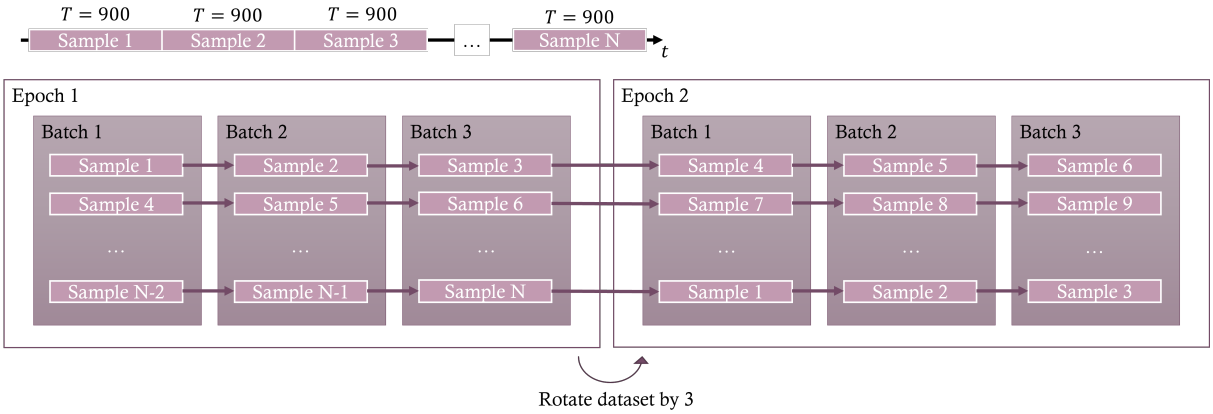


Figure 4.3: Visualisation of batch processing during training. The original time series (top) is divided into even duration samples of 900 seconds. These are presented to the model in batches during training, where the membrane potentials are kept between batches and epochs. Model parameters are updated after each batch. At each epoch, the dataset is rotated so that the temporal order of samples is kept (e.g. first sample of the first batch in epoch 2 is follows the first sample of the last batch of epoch 1). In this example, there are three batches, so that the dataset is rotated by 3. Hence, sample 4 is the first sample, initialised with the membrane potential of running sample 3 last.

A prediction for a sample is made depending on the maximum membrane potential at the output layer at each timestep Δt. This allows the use of regular loss functions for optimisation. Similar to Zenke et al. (2018) [28], the models are optimized on the negative log-likelihood loss, which is appropriate for multi-class probabilistic models.

Even though the main focus of this work is not the optimisation of SNNs, the underlying models to explanation research should demonstrate a clear improvement in performance to pure chance, so that an explanation likely represents what the network has learned. Therefore, the hyperparameters of the networks are tuned in a greedy optimisation process under the assumption of independence. This means that initial hyperparameter values are set so that one after the other can be tuned. Once the best option is determined for one hyperparameter, the initial value for tuning the next is replaced. This greedy optimisation was mainly done to increase model building speed. Each hyperparameter was tuned by training an SNN with 20 epochs on the training set and evaluating it on the validation set after training. The lowest validation loss determines the optimal hyperparameter. The full list of hyperparameters and the detailed tuning results can be found in appendix B. Only the optimiser for adapted gradient descent learning has been fixed to Adam, based on preliminary experiments.

Hyperparameter	**OneLayerSNN**	**TwoLayerSNN**	**ThreeLayerSNN**
Δt	0.001	0.001	0.001
τ_{syn}	0.01	0.01	0.01
τ_{mem}	0.01	0.001	0.01
Learning rate	0.01	0.001	0.001
Batch size	128	256	512
Size of hidden layer 1	-	100	50
Size of hidden layer 2	-	-	25

Table 4.2: Results of the hyperparameter tuning for all models.

4.2.3 Final Models

With the hyperparameters in Table 4.2, the final models were fully retrained on the training set. As a regularisation, early stopping with a patience of 10 epochs was used, monitoring the validation loss. To evaluate the models, the balanced accuracy is reported on the test set at a 95% confidence interval (CI) (Table 4.3).

The CI was calculated using the following formula. Let n be the cardinality of the respective datasets (i.e. train, validation, test).

$$\text{CI}_{0.05} = 1.96 \times \sqrt{\frac{\text{Balanced Accuracy} \times (1 - \text{Balanced Accuracy})}{n}} \tag{4.1}$$

Balanced Accuracy

Balanced accuracy is a performance metric for supervised classification tasks [51]. Unlike the common accuracy metric, balanced accuracy considers class imbalance and is more robust in its indication toward model performance concerning imbalance. Formally, balanced accuracy is the arithmetic mean of sensitivity and specificity in a binary classification case. For multi-class classification, there exist different definitions. The definition used in the frame of this work follows Mosley [52], who defined it as below:

$$\text{Balanced Accuracy} = \frac{1}{N} \sum_{C} \min(P_C, R_C) \tag{4.2}$$

where N is the number of classes in the dataset, P_C the precision and R_C the recall of class C. In this work, the implementation of Python's scikit-learn library is used [53].

Model	**Balanced Accuracy**		
	Test	**Train**	**Val**
OneLayerSNN	0.516 ± 0.001	0.506 ± 0.001	0.536 ± 0.001
TwoLayerSNN	$\mathbf{0.517 \pm 0.001}$	$\mathbf{0.515 \pm 0.001}$	$\mathbf{0.549 \pm 0.001}$
ThreeLayerSNN	0.500 ± 0.001	0.490 ± 0.001	0.520 ± 0.001

Table 4.3: Balanced Accuracies of the SNN models reported at 95% CI.

The model performances (Table 4.3) show that the SNN models used for this thesis do not outperform recent work on this dataset (e.g. Hamad et a. (2021) [54] who used a convolutional neural network with dilated causal convolution and self-attention and achieved F1-scores of 90.78 and 87.34 for the activity prediction task per subject A and B respectively). Nevertheless, the models perform significantly better than chance on an 11-class classification problem. Therefore, they are suitable for research into local explanations for SNNs. Interestingly, the performances of all models are quite similar, with *TwoLayerSNN* having the best performance and the *ThreeLayerSNN* having the worst on all sets. The inference models act as the use case models to carry out further research on extracting explanations from SNNs using the algorithm defined in chapter 5.

5 TEMPORAL SPIKE ATTRIBUTION EXPLANATIONS

Little explanation methods for predictions with spiking neural networks exist. In the frame of this thesis, a method is developed to provide a local, feature-based explanation, which is inspired by attribution methods for ANNs. This chapter defines feature attribution and presents *Temporal Spike Attribution* (TSA), an algorithm to compute feature attribution in SNNs, demonstrated on the models presented in chapter 4.

5.1 Feature Attribution Definition

There are several names for explanations that provide interpretable knowledge through relevant parts of the input to an output [36]. Saliency, feature relevance, contribution, and attribution are some, for example. In this work, the term *feature attribution* shall be used.

Then, each input feature has an attribution value that indicates how much it attributes to a single network prediction. In the case of time series data, a feature is not solely defined by the input dimension, but also by the time step of the input.

Formal Definition of Feature Attribution for Time Series Data

Let $x \in \mathbb{R}^{D \times T}$ denote a single multivariate time series within a time series dataset. D describes the cardinality of the input in terms of input dimensions (e.g. number of sensors in the case of the ADL dataset). Consequently, T is the duration of x. Let O be the cardinality of the output, i.e., the number of output classes.

Then, the feature attribution $A^{O \times D \times t}(x, t)$ is defined for each input dimension at each time step up until current time $t \leq T$ of x to each output dimension o, where each combination of input dimension and time step computes an attribution value $a_{o,d,t}(x, t)$.

$$A^{O \times D \times t}(x,t) = \left(\begin{pmatrix} a_{1,1,1} & a_{1,1,2} & \cdots & a_{1,1,t} \\ a_{1,2,1} & a_{1,2,2} & \cdots & a_{1,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,D,1} & a_{1,D,2} & \cdots & a_{1,D,t} \end{pmatrix} \quad \cdots \quad \begin{pmatrix} a_{O,1,1} & a_{O,1,2} & \cdots & a_{O,1,t} \\ a_{O,2,1} & a_{O,2,2} & \cdots & a_{O,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{O,D,1} & a_{O,D,2} & \cdots & a_{O,D,t} \end{pmatrix} \right) \tag{5.1}$$

These attribution values indicate the past time series' impact on the current prediction at the current time step t, where a positive value indicates a positive relationship (e.g. an active bed sensor could be positively related to the prediction of *Sleeping*) and a negative value shows a negative relationship respectively (e.g. an active door sensor could contribute negatively to the prediction of *Sleeping*).

The explainability of the models is provided through the visualisation of the feature attribution map which results from TSA. The following sections detail the computation of the attribution.

5.2 Temporal Spike Attribution Components

As related work [45, 46] has shown, there are multiple methods to arrive at a feature attribution explanation for time series data in ANNs. Due to the difference in information processing between SNNs and ANNs, gradient-based approaches which are widespread for ANNs [36, 55], are not applicable. Kim and Panda (2021) [40] showed that even for SNNs that were trained with surrogate gradient learning, gradient-based feature attribution explanations are not optimal, as the gradients are smoothed with a surrogate. Therefore, the feature attributions become smoothed as well, leading to an unclear attribution map where every feature is somewhat relevant to the prediction. Thus, a new feature attribution computation for SNNs is required. This research takes the approach of a model-agnostic method for temporally coded SNNs, considering the different model-internal information available at model inference.

An SNN processes spike trains across several layers to arrive at a prediction. Therefore, the information that is available in one prediction of one input x is the following: (1) the exact spike times of every neuron of every layer $S^{(l)}$ (Section 5.2.1), (2) the learned weights of the network W (Section 5.2.2), and (3) the membrane potential of the neurons at the output layer $U^{(L)}$ (Section 5.2.3). Each of these has a certain relationship to the prediction, which is elaborated below. By combining them, a formula for feature attribution of temporally coded SNNs is derived.

5.2.1 Influence of Spike Times

In temporal coding, the information about the stimulus, or data, is assumed to be in the exact firing times of a neuron [17]. The exact spike times of each neuron indicate the attribution of neurons to their downstream neurons, which includes overall feature attribution information to the prediction at the time of the explanation t. Therefore, the spike times are to be analysed in relation to t.

A LIF neuron i fires a spike if its membrane potential u_i crosses a certain threshold θ [17]. As u_i decays exponentially over time, frequent input spikes are more likely to stimulate the neuron to fire, as they impact the membrane potential additively. Consequently, sparse input spikes are less likely to generate output spikes. With regards to t, this means that recent input spikes attribute more to the state of the neuron than input spikes that lie further in the past. To capture this relationship between the exact spike times and their attribution, Kim and Panda (2021) [40] introduce the neuronal contribution score (NCS) (see section 3.1.2). The NCS sums the temporal spike contribution scores (TSCS) of one feature's spike train (e.g. a pixel in an image). The TSCS describes the contribution of a single spike in a spike train to the current time t. Hence, the NCS represents the contribution of input spikes of a neuron to its downstream neurons.

The model prediction is based on the membrane potential of the output layer. Applied to the output layer of an SNN, the NCS' of the hidden layer neurons indicate the attribution of hidden layer spikes to an output neuron's membrane potential as it is directly connected. Therefore, the NCS of all layers before the output layer consolidates the exact spike times' effect of the network on the prediction at each time step and should be considered in the feature attribution computation.

Formally, the contribution of a neuron i to the prediction at current time t in the model is modelled through the NCS $N_i(t)$ (see equation (3.5) in section 3.1.2). This score is characterised by γ, which specifies the steepness of the exponential decay over time. To reflect the dynamics of the model in the explanation, we define the decay at the same rate as the decay of the LIF neuron's membrane potential, which generates spikes if threshold θ is crossed.

$$\gamma = \frac{\Delta t}{\tau_{mem}} \tag{5.2}$$

In the case of a multivariate binary time series dataset, such as the ADL, with a prediction task at each time step, the NCS can be slightly simplified. The NCS can at most consider one spike at t' with regard to explanation time $t > t'$, meaning that the summation in the NCS computation can be neglected.

If there is no spike at t', the neuron i does not contribute actively to a change in membrane potentials of the downstream neurons. Instead, the effect of a non-spike can be interpreted in two possible ways. On the one hand, no effect can be understood as such, which means that the NCS of neuron i would be 0 at the time of the non-spike. On the other hand, no spike can be understood to contribute to the downstream neurons by not increasing or decreasing their membrane potential by the postsynaptic potential. Therefore, the *spike time ingredient* $N_{i,t'}(t)$ is defined per neuron i of the model and exhibits two possible definitions:

$$N_{i,t'}(t) = \begin{cases} \exp(-\gamma|t-t'|) & \text{if } x_{i,t'} = 1 \\ 0 & \text{Otherwise} \end{cases} \tag{5.3}$$

$$N_{i,t'}(t) = \begin{cases} \exp(-\gamma|t-t'|) & \text{if } x_{i,t'} = 1 \\ -\exp(-\gamma|t-t'|) & \text{Otherwise} \end{cases} \tag{5.4}$$

Then, the NCS with regards to t can be computed for each layer and time step before t, resulting in vectors $\vec{N}^{(l)}(t)$ that have the same size of the respective layer. Let n be the size of layer l:

$$\vec{N}^{(l)}_{t'}(t) = \langle N_{1,t'}(t), N_{2,t'}(t), ..., N_{n,t'}(t) \rangle \tag{5.5}$$

5.2.2 Influence of Model Parameters

The network parameters that are learned during the training process are referred to as the network's learned weights. In this research, static weights are considered because they are more common, unlike the time-varying weights that form the basis for Jeyasothy et al. (2019)'s work [38]. Hence, the weights influence the attribution of an input to the network's prediction.

The weight values W represent the strength of the synapses in a neural network. In SNNs, they determine the exact postsynaptic potential as the input values are always 1 in case of a spike, and 0 if there is no spike at a time t. Therefore, the weights determine the impact on the postsynaptic neuron's membrane potential directly, where the absolute weight value indicates the weight's attribution to the postsynaptic neuron.

Moreover, the sign of the weight specifies whether the synapse is excitatory or inhibitory. The nature of the synapse influences the change in membrane potential of the postsynaptic neuron, where positive weights increase and negative weights decrease the potential. Consequently, the sign of the weight matters in identifying the spikes that caused a neuron to spike.

In summary, the computation for feature attribution in SNNs shall consider both the value of a weight as well as its sign. To keep the effects of the weight values relative, the absolute of the weight matrix $|W^{(l)}|$ connecting two layers are normalised with a min-max normalisation. The absolute is taken as the absolute values dictate the effect of the synapse (i.e., the synapse strength). Additionally, to keep the excitatory or inhibitory nature of the synapse, the signs are considered:

$$C_W(W^{(l)}) = \mathsf{sign}(W^{(l)}) \circ \frac{|W^{(l)}| - \mathsf{min}(|W^{(l)}|)}{\mathsf{max}(|W^{(l)}|) - \mathsf{min}(|W^{(l)}|)} \tag{5.6}$$

where $\circ$ denotes an element-wise matrix multiplication. So, C_W is also a two-dimensional matrix with the same dimensions as W. It is noteworthy that the weight attribution is the same for each input and constant across time because it is a property of the inference models. Thus, this attribution gains its meaning from the combination with the other components.

5.2.3 Influence of the Output Layer's Membrane Potential

The output layer is the last computational layer in an SNN. The activity of the output layer is the basis for the prediction, thus it influences the model prediction. Hence, the output layer component should be considered in feature attribution computation. The output layer consists of spiking neurons that contain the state variables u (membrane potential) and s (spike train). The prediction is done based on the output layer's membrane potential u. This is the reason why this component is to be considered in the computation of feature attribution. The spike trains s are neglected as a consequence[11].

This component gains its meaning from the relative states of the other output neurons. High membrane potential is connected to high classification confidence only if the other output neurons exhibit lower potentials. Therefore it is important to consider the relative values in the output layer. Through the membrane potential, the certainty of the model's prediction can be accessed, which is also referred to as prediction confidence. Additionally, it is a piece of interesting information for the explanation overall.

The classification confidence $P_i(t)$ of output neuron i for current time t is defined as the softmax probability of the output membrane potentials at the output layer in this work. In this way, the membrane potentials are normalised in a fixed interval and represent the class probabilities which are interpreted as classification confidence. Let L denote the set of output neurons so that $i \in L$. Let j be a neuron $\in L$. Output neuron i's confidence is then computed as such:

$$P_i(t) = \frac{\exp u_i(t)}{\sum_j \exp u_j(t)} \tag{5.7}$$

Similar to the NCS, the classification confidence can also be computed for the whole output layer at once. Then, the confidence values are specified in the confidence vector $\vec{P}$, where O is the size of the output layer:

$$\vec{P}(t) = \langle P_1(t), P_2(t), ..., P_O(t) \rangle \tag{5.8}$$

5.3 Temporal Spike Attribution Formula

Considering the variables discussed in the previous section as components, a formula can be derived with a summation and multiplication approach for the calculation of a feature attribution score. This formula is referred to as *Temporal Spike Attribution* (TSA).

In this thesis, a forward approach taking the work of Kim and Panda (2021) [40] as inspiration is followed. This means that the final attribution scores are retrieved by aggregating the attribution

[11] In the case of a SNN that predicts based on earliest spikes, the NCS of the output layer should be considered instead of the membrane potential.

elements (i.e., spike times, weights, output membrane potential) in the input domain. This is done class-wise so that the final results are feature attributions of the input to each output class.

A neuron $i^{(l)}$ generates the spike train s_i to the downstream computational layers. It is fully connected via synapses to the next layer given the SNN models. Therefore each neuron is connected to all neurons of the following layer $l+1$ with the weight matrix $W^{(l)}$.

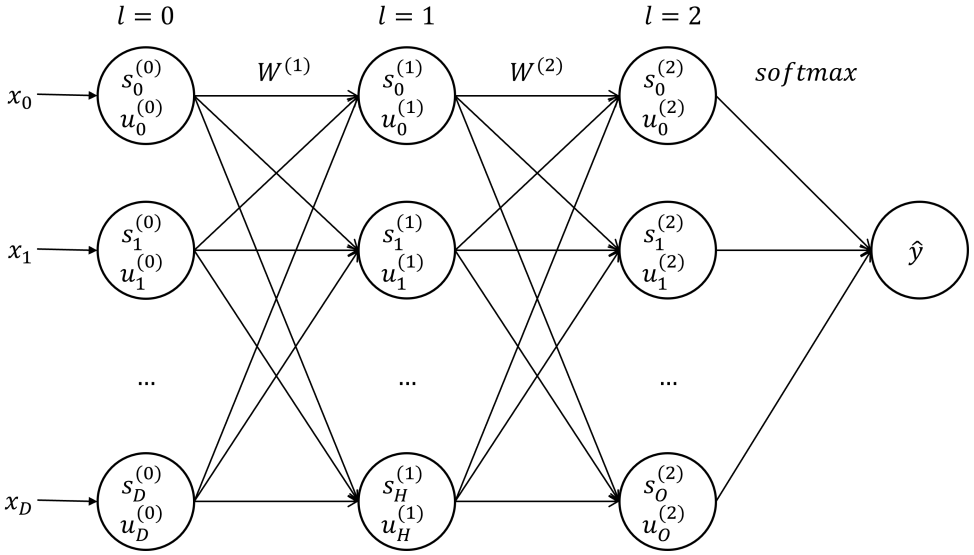


Figure 5.1: Information available for the computation of *Temporal Spike Attribution* of *TwoLayerSNN*: Available internal states when presented with input $x \in \mathbb{R}^{DxT}$. l denotes the layer, $s_i^{(l)}$ the spike train of neuron i of layer l, $u_i^{(l)}$ the membrane potential of i at l respectively. $W^{(l)}$ marks the weight matrix of l and $\hat{y}$ is the predicted class.

The NCS $\vec{N}^{(l)}(t)$ represents the spike times, the weight contribution C_W represents the weights and the classification confidence $\vec{P}(t)$ represents the output layer membrane potentials as components of the Temporal Spike Attribution algorithm (see Figure 5.2).

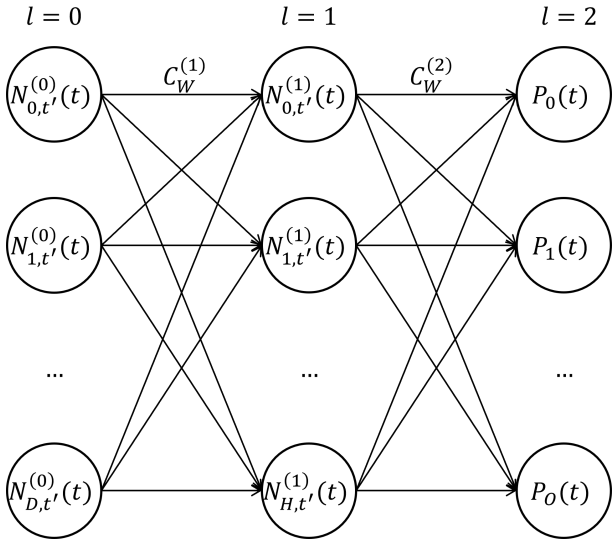


Figure 5.2: Components for *Temporal Spike Attribution* of *TwoLayerSNN* at t' with regards to t: $N_{i,t'}^{(l)}$ is the NCS of neuron i for time step $t' < t$ with regards to explanation time t at layer l. $C_W^{(l)}$ represents the weight contributions, and $P_i(t)$ denotes the classification confidence for class i.

The first two are combined by multiplying the diagonal matrix of $\vec{N}^{(l)}(t_c)$ with C_W. This operation

results in a weighted NCS matrix $N_W^{(l)} \in \mathbb{R}^{n \times m}$, where n is the size of layer l and m is the size of the next layer $l+1$. The weighted NCS matrix can be computed for all layers, except for the output layer. The result is a matrix consisting of scores for each synapse within the network, representing how the presynaptic neuron contributes to the postsynaptic neuron under direct consideration of the synapse weight.

$$N_W^{(l)}(t) = \text{diag}(\vec{N}^{(l)}(t)) \cdot C_W^{(l)} \tag{5.9}$$

To aggregate these values across layers, a multiplication and summation approach is taken. Since each input neuron is connected to each output neuron through multiple paths, the values are summed to arrive at a single value for each feature at a given time. If there are hidden layers in the network, the weighted NCS values that are connected across the layers in depth are multiplied (Figure 5.3). This addition and multiplication process can be described through multiple matrix multiplications of the weighted NCS' of the different layers, simulating a forward pass of the model. This represents how the input influences the neurons of the network. The final feature attribution $A(x,t) \in \mathbb{R}^{O \times D \times t}$ is computed through multiplication with the classification confidences, as shown in algorithm 1.

As there are two different interpretations of non-spiking attribution, *Temporal Spike Attribution - Spikes only* (TSA-S) which uses (5.3) as the definition for $N_{i,t'}^{(l)}(t)$ and *Temporal Spike Attribution - Non Spikes included* (TSA-NS) with (5.4) as the definition for $N_{i,t'}^{(l)}(t)$ are distinguished in this thesis.

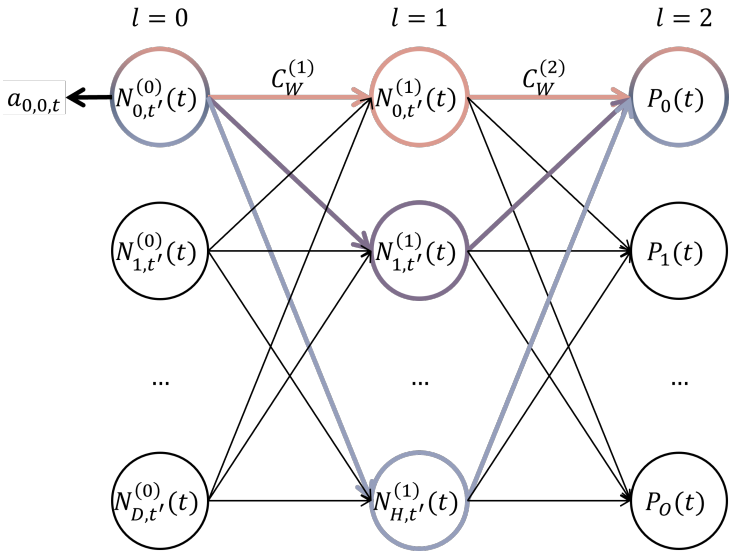


Figure 5.3: Visualisation of the computation of $a_{0,0,t}$ of $A^{D \times O \times t}$ using *Temporal Spike Attribution*. The neurons highlighted through colour are part of the computation, where one colour corresponds to the multiplication of these elements. The different products are then added to map the input to the output. Here, $N_{0,t'}^{(0)}(t)$ and $P_0(t)$ are used multiple times in the computation.

Algorithm 1 Temporal Spike Attribution

Let x be an input in $\mathbb{R}^{D \times T}$, f the SNN model with L layers, $S^{(l)}$ the spike trains of layer l, $U^{(L)}$ the membrane potential of the output layer, and t the current time.

$S^{(1)}, ..., S^{(L-1)}, U^{(L)} \leftarrow f(x)$ ▷ Run the input and retrieve internal variables.
$\vec{P}(t) \leftarrow \text{softmax}(U^{(L)})$
for $l = 1, 2, ..., L$ **do**
  $C_W^{(l)} \leftarrow \text{sign}(W^{(l)}) \circ \frac{W^{(l)} - \min(W)}{[\max(W) - \min(W)]}$
end for
for $t' = 0, 1, 2, ..., t$ **do**
  Initialize $F(t) = I \in \mathbb{R}^{DxD}$
  for $l = 1, 2, ..., L-1$ **do**
    if $S^{(l)}(t) = 1$ **then** ▷ In case there was a spike at t'
      $\vec{N}^{(l)}(t) \leftarrow \langle N_1(t), N_2(t), ..., N_n(t) \rangle$ where $N_{i,t'}(t) = \exp(-\gamma|t - t'|)$
    else ▷ In case there was no spike at t'
      $\vec{N}^{(l)}(t) \leftarrow \langle N_1(t), N_2(t), ..., N_n(t) \rangle$ where $N_{i,t'}(t) = 0$ (TSA-S)
        or $N_{i,t'}(t) = -\exp(-\gamma|t - t'|)$ (TSA-NS)
    end if
    $N_W^{(l)}(t) \leftarrow \text{diag}[\vec{N}^{(l)}(t)] \cdot C_W^{(l)}$
    $F(t) \leftarrow F(t) \cdot N_W^{(l)}(t)$
  end for
  $A(x,t) \leftarrow F(t) \cdot \text{diag}(\vec{P}(t))$
  Concatenate $A(x,t)$ to feature attribution map $A(x,t)$.
end for

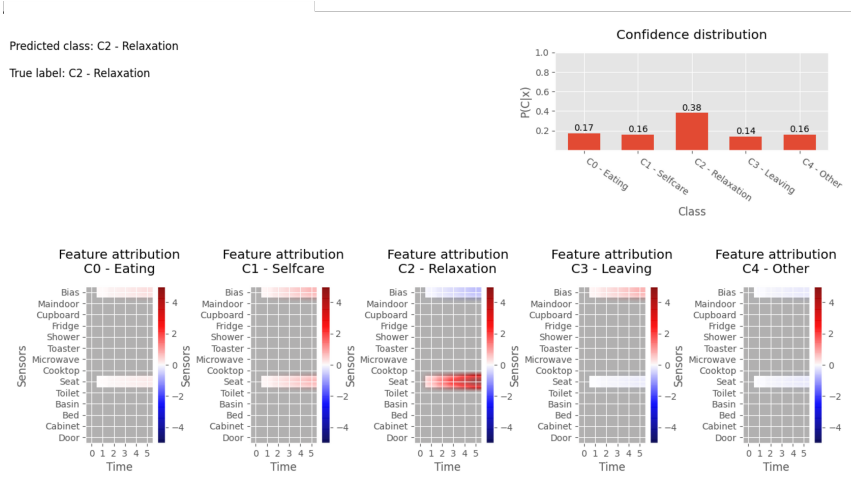
5.4 Visualisation

To present the information in the attribution map as a local explanation to the target group of model developers, the attributions need to be visualised. Additionally, the information of model prediction, ground-truth label, and classification confidence is provided in the visualised explanation. The design of the visualisation is defined through three short design iteration cycles with TSA-S explanations within the research team of this thesis. We emphasise that this process does not replace a separate study regarding optimal visualisation. Instead, it shall offer sufficiently good visualisation of the attributions computed from TSA which can be used in the evaluation. This section gives a short overview of the design process.

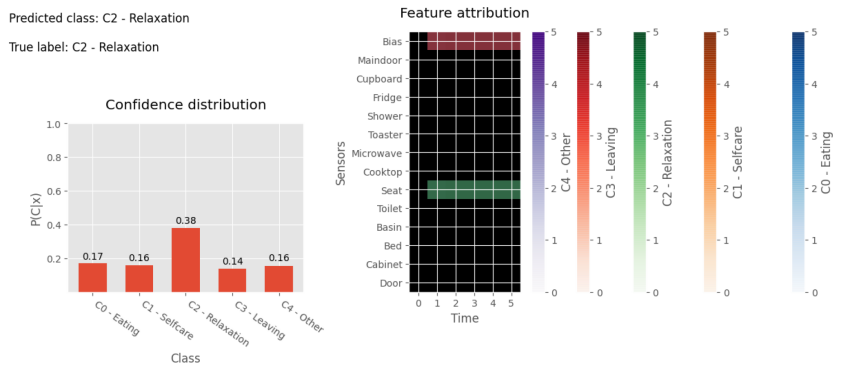
5.4.1 First Iteration - Initial Visualisation

The resulting feature attribution map retrieved from TSA is a three-dimensional map that details the attributions of each input dimension d at each timestep prior to the prediction at timestep t to each output class o. Three dimensions are not straightforward to display on two-dimensional surfaces such as a screen. Therefore, two options for visualisation are explored in the first iteration: (1) Visualisation of all two-dimensional maps per class (Figure 5.4a), (2) Collapse the three-dimensional map into one map (Figure 5.4b). In the latter option, the class that a feature attributed to the most is shown in the visualisation. The classification confidence distribution is presented as a bar graph. Moreover, the predicted class and ground truth are given as text information.

In the first approach, the attributions per class can be distinguished between negative and positive, visualised through a diverging colour map from blue to red. It offers more information as



(a) Approach 1: Separated attribution maps per class.



(b) Approach 2: Collapsed two-dimensional map.

Figure 5.4: First iteration of visualisation design for TSA using TSA-S explanations. The first design iteration was conducted on preliminary experiments with SNN models trained on a five-class classification problem.

the different class attributions can be compared. However, the amount of information can also be overwhelming, especially in a multiclass case such as the ADL from binary sensors task. In contrast, the second approach only shows one feature attribution map, where the different class attributions are highlighted in different colours. As the second approach is better suited for multiclass problems, we decided on a collapsed map in the visualisation.

As shown in Figure 5.4b, sensor activation is shown in white, while the absence of spikes is indicated in black. The class attribution of a feature is overlaid with a certain degree of transparency. The meaning of the spikes and non-spikes in the data did not become clear in this first iteration. Moreover, this approach does not suit TSA-NS explanations, as a coloured overlay to black (i.e. non-spikes), is not visible. Therefore, the data visualisation needs improvement to clarify the spiking data and allow for a clear attribution highlight.

Furthermore, the class colours used for the feature attribution map are each indicated as their own colour bar next to the map, which requires a lot of space with multiple classes. Besides, the colours were chosen from the available built-in colours of the plotting library [56]. Due to the number of classes, certain colours are not easily distinguishable (e.g., red and orange). Thus, the colour scheme requires improvement as well as the visualisation of the colour to class mapping. Additionally, the correspondence of the class confidence distribution to the attribution map was not found to be clear instantly. An alignment with the other colours used would be appropriate.

5.4.2 Second Iteration - Spikes and Colours

In the second iteration, improvements were made in the visualisation, specifically concerning the data visualisation and colour scheme. The data is visualised per spike where each spike is a vertical black line. The background is white, thus non-spikes are also white, which enables a clear view of the attribution of non-spikes, too. A colour scheme with high perceptual distance is chosen using Brown University's Cologorical tool [57] to represent the class colours. The colour scheme is also adapted in all parts of the visualised explanation (i.e., true label and predicted label text, confidence distribution, feature attribution map) to enable immediate categorisation and provide consistency of the displayed classes. Besides, the colour bars next to the feature attribution map are replaced with a legend of the colour scheme used throughout the whole explanation. Another small alteration is the exclusion of the class numbers (e.g., "C1") in front of the class labels, as this is not needed for the user. Furthermore, the confidence distribution part of the visualisation shows the development of the model's classification confidence over time up until the prediction at t. This should give some insight into the model behaviour up until t to the user. It was found that the feature attributions for all SNN models of the use case are largely covered in the presentation of the last 60 seconds prior to prediction time t. Hence, the explanation covers the time frame of $[t-60, t]$ in the visualisation.

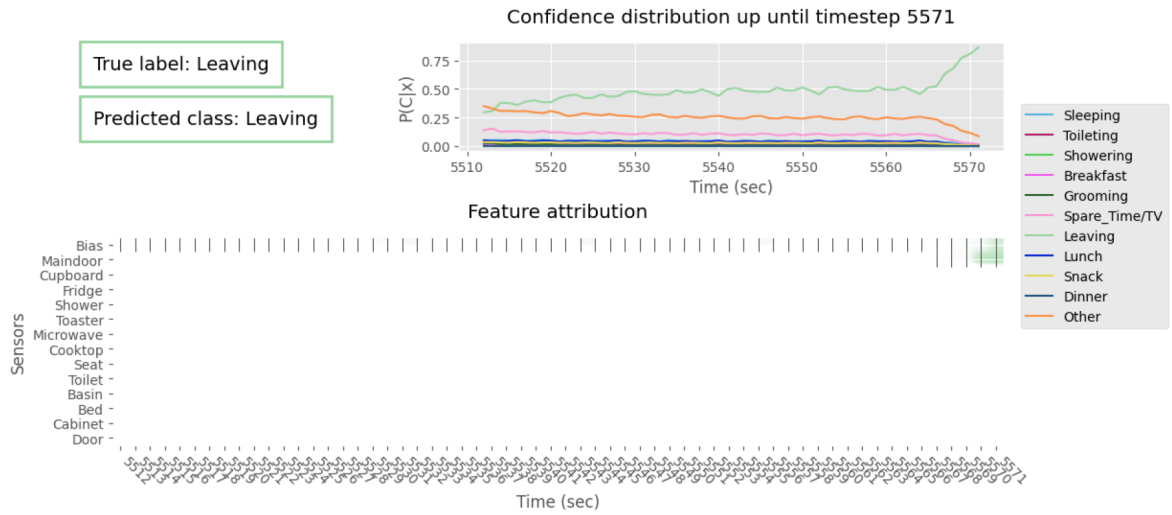


Figure 5.5: Second iteration of visualisation design for TSA using TSA-S explanations.

While the second iteration improved on the first version, there is one main concern. Although the development of the confidence distribution over time is interesting, it can confuse the user. The feature attribution map shown relates each feature attribution to the prediction at time t. The confidence distribution, in contrast, shows the model's confidence at each time $t' < t$. Therefore, the possibility to misinterpret the feature attributions of features at $t' < t$ to correspond to the confidence development over time exists. Hence, clarity could be improved.

5.4.3 Third Iteration - Confidence

The third iteration of the explanation visualisation design yields the final visualisation (Figure 5.6). The confidence distribution is switched back to the bar plot of iteration one in order to emphasise the validity of the explanation for prediction at time t. This visualisation offers dense information about the feature attributions and confidence distribution of the model, with consistency in the different parts of the visualisation through a colour scheme with high perceptual

distance.

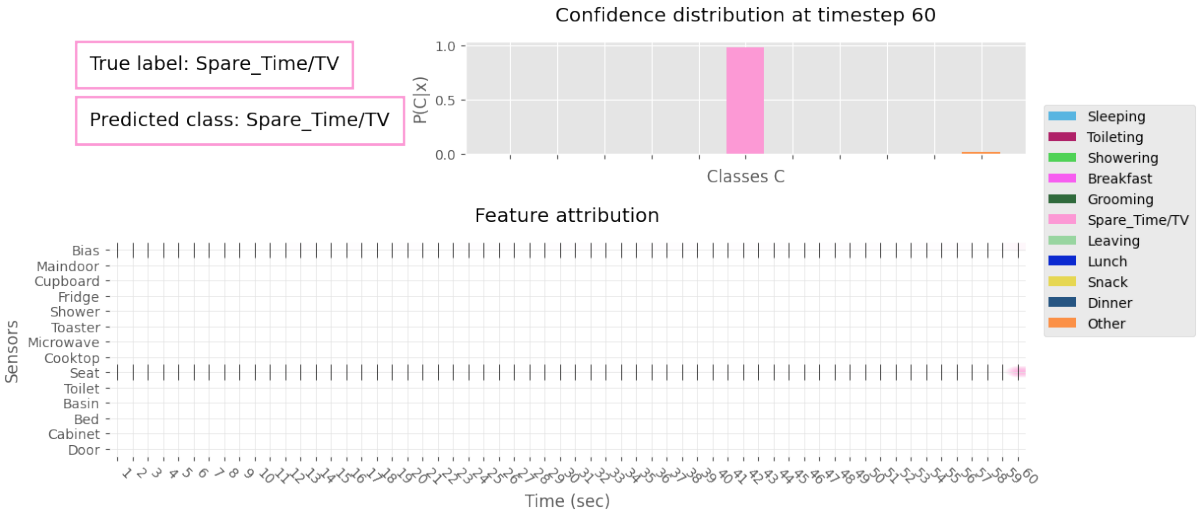


Figure 5.6: Third and final iteration of visualisation design for TSA using TSA-S explanations.

The design of the explanation visualisation is determined using TSA-S explanations, for which the visualisation makes sense and showed coherent results in the eyes of the researcher. However, the TSA-NS explanations do not seem to make sense when collapsed into one two-dimensional map (Figure 5.7). The sensor activation of the data is not attributing to the predicted class in TSA-NS, and the found feature attributions do not seem to make much sense when consolidating the three-dimensional attribution map into one from the researcher's perspective. Hence, a need for revising the weight of the non-spiking attribution is required to improve TSA-NS in this regard before conducting user studies to measure human-comprehensibility.

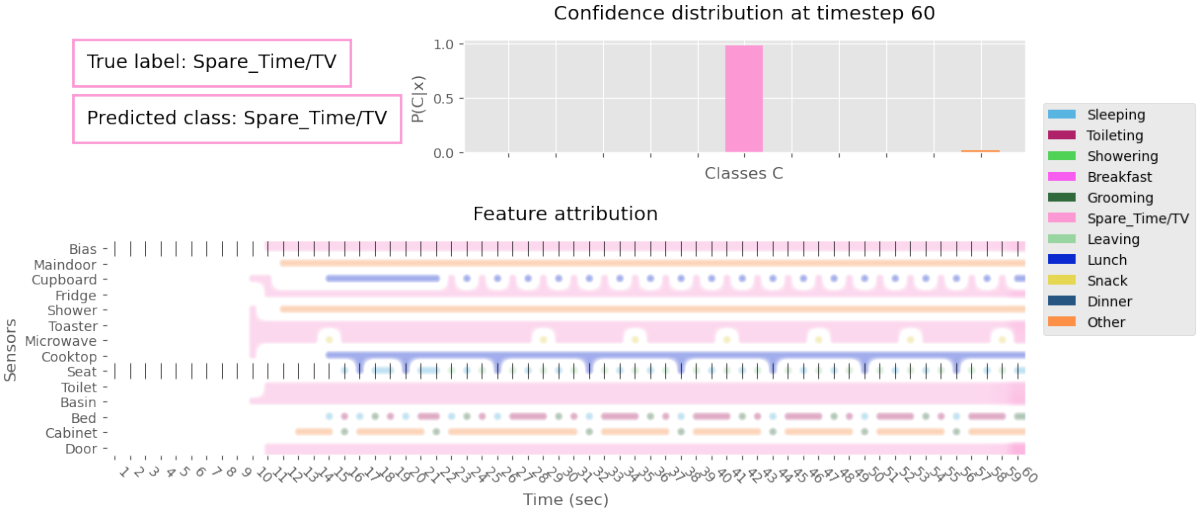


Figure 5.7: Visualisation of a TSA-NS explanation.

As the TSA-NS explanation visualised in this manner showed major incoherence between the data and the ground truth as well as model prediction, only the TSA-S explanations are evaluated visually in the frame of this research.

6 EXPLANATION QUALITIES

The preceding chapter 5 detailed how a feature attribution explanation is achieved for SNN models using *Temporal Spike Attribution* (TSA). This offers a local explanation of the model's predictions. To assess the quality of the explanation, it needs to be measured and evaluated. Unlike prediction performance, explainability does not have a standard set of measures, as the requirements for a good explanation are not straightforward and highly target-group dependent [6]. Previous works in XAI for SNNs have focused on validating the explanation reliability of interpretable knowledge [37] or the accuracy of the explanation, considering another explanation as ground truth [40]. Both did not aim at providing a thorough evaluation procedure for interpretability but demonstrated the functionality of their explanation methods. Moreover, these qualities are mainly related to whether the explanation is truthful, however, there are more aspects to a good explanation to be considered. Therefore, this chapter discusses qualities to measure TSA by with respective concrete metrics as well as the experimental results using the models and task proposed in chapter 4. The evaluation is split into a technical evaluation based on faithfulness [42], stability, robustness, certainty, and a user evaluation for human-comprehensibility [36, 48]. These qualities have been identified from related work in XAI with time series tasks, as presented in chapter 3.

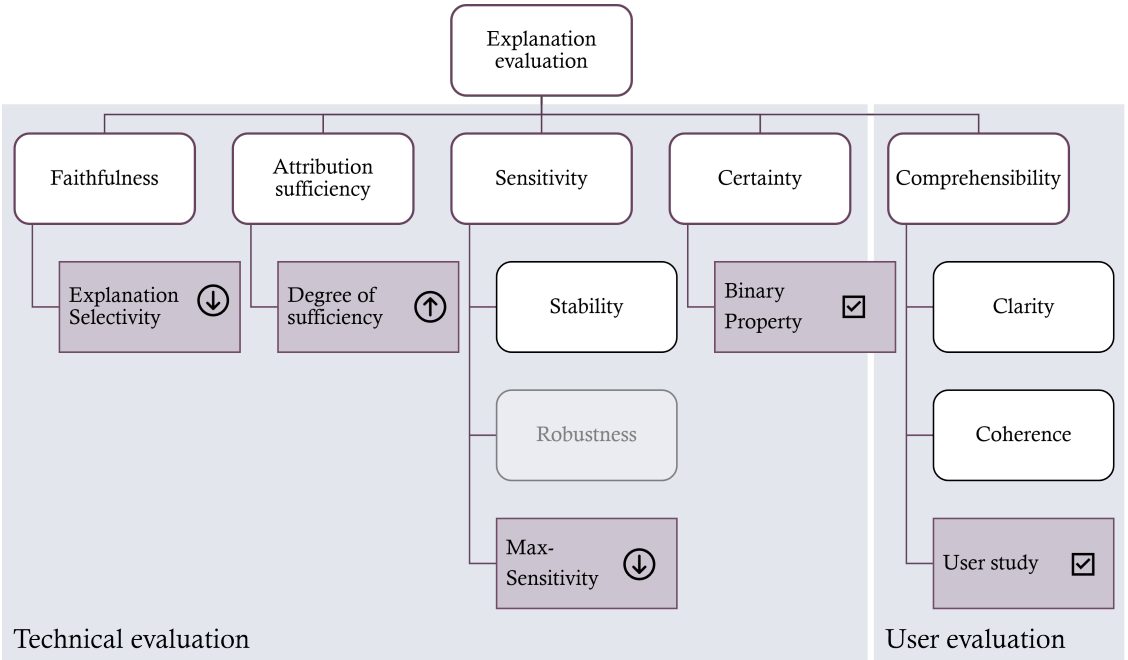


Figure 6.1: Evaluation framework for local feature-based explanations on time series data with qualities and metrics. Arrows indicate optimal values of the metrics, check marks indicate desired fulfilment of property. In this thesis, robustness is not evaluated.

6.1 Technical evaluation

The technical evaluation inspects the quality of TSA explanations in a functionally-grounded evaluation [58]. Such an evaluation is appropriate because TSA as an explanation method is not yet validated in its explainability. Through a technical evaluation that does not require human evaluation, a fundamental evaluation of TSA's effectiveness and functionality as a local explanation method for temporally coded SNNs is performed. Hence, the technical evaluation concerns all quality aspects which are not directly linked to human understanding and can therefore be evaluated without humans. Namely, these are (1) TSA's faithfulness to the model behaviour, (2) the sufficiency of the feature attributions for the model prediction, (3) TSA's sensitivity to similar data, and (4) TSA's certainty in the explanation. In this section, the experimental setup, as well as the results of the experiments, are presented and discussed. While the first three qualities are elaborated further in this section, certainty is a binary quality that does not require a dedicated evaluation and can therefore be discussed beforehand.

Certainty

The *certainty* [36] of an explanation is a quality that indicates whether the explanation informs the user about the confidence of the model in its prediction. Rojat et al. (2021) [48] identify this information as a fundamental part of a trustworthy explanation method. By quantifying how confident the model is in its prediction, the explanation is put into context. Users know how sure the model is, and can therefore judge the model prediction under that consideration. Certainty contributes to the transparency of the explanation and hence is a desirable quality. In the frame of this thesis, the definition of certainty is restricted to the underlying model's prediction confidence and is not referring to the explanation method's confidence in the quality of the explanation.

As certainty is binary, it is not measurable by evaluation. Instead, it is a property that an explanation method fulfils or not. In the case of the explanation method presented in this thesis, the classification probability $P(C|x)$ is interpreted as the classification confidence, or *certainty*. It is part of the explanation, hence the explanation method provides *certainty*.

6.1.1 Experimental Setup

The technical evaluation is based on a specific test set selected for evaluation of TSA's explainability due to computational efficiency. This section explains the selection of the test set for technical evaluation first before presenting the tested metrics and experimental procedures.

Test Data

As a basis of the technical evaluation of the explanations, test set data from model training is appropriate since it represents unseen data to the SNN models. Unfortunately, the complete assessment of TSA is not feasible on all test data due to the non-optimal efficiency of the SNN model implementation in terms of runtime and memory efficiency: When extracting explanations, the whole time series from $t = 0$ until the respective timestamp is considered. As state variables (i.e., membrane potential and spike train) are retained from start to finish of a simulation and complete records of the state variables of this time are required for the explanation, the data has to be processed sequentially. This leads to slow processing as well as large memory requirements due to the large size of the state variable records.

Therefore, the test set is sub-sampled to select evaluation data for the technical evaluation (see Figure 6.2). For each time series of subjects A and B of the dataset, nine timestamps are chosen per class that is present in the test set. From these timestamps per class, three are required to be each from the beginning, middle and end of the respective activity to ensure that different temporal characteristics and changes of activities are present in the data used for evaluation. The beginning and end of the activity are defined as the first and last minute respectively. Given these constraints, the timestamps are sampled at random. This results in a total of 180 timestamps across the test set (i.e., 81 for data of subject A and 99 for data of subject B) for which explanations are extracted and evaluated with regards to the qualities presented in this chapter. Choosing the same number of timestamps per class additionally balances the evaluation of TSA. Furthermore, an assumption is made to limit the length of the time series presented to the network. It is assumed that only the information of the last hour prior to the timestamp of the explanation includes information that is relevant to the model prediction. Therefore, the explanations for the evaluation are extracted for the timeframe of one hour prior.

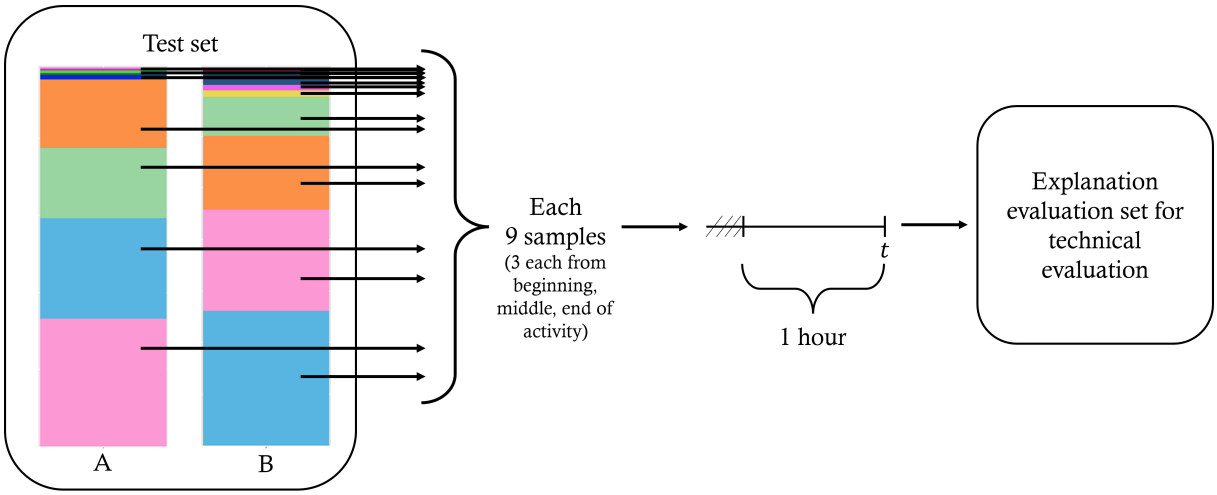


Figure 6.2: Sampling process for the dataset for technical evaluation. From each subject data A and B, nine samples are collected per class of the test set that includes three samples from the beginning, middle and end of an activity respectively. Colours represent classes. Then, all samples are shortened to a maximum of one hour.

Since there is limited related work in local explanations for temporally coded SNNs, there is no reported baseline to compare to. Therefore, a baseline is generated through the assignment of random attribution scores in the extracted test set explanations. The baseline assumes that all spikes in the input exhibit an attribution value. Therefore, random attribution values in the interval between the minimum and maximum recorded attribution value of the extracted explanations are assigned to the spiking parts of the data.

Feature Segments

TSA generates feature attribution explanations. Hence, the feature attributions are the subject of the technical evaluation. The feature attributions are computed per input dimension and time step. The attributions, however, cannot be interpreted at the granularity of one input dimension and time step because of the temporal dependencies (e.g., the attribution of the *Bed* sensor at $t = 50$ to the class *Sleeping* is particularly large not only because the sensor is activated at $t = 50$, but also because the sensor is activated in the previous time steps). Therefore, the technical evaluation is based on so-called *feature segments*. In the frame of this thesis, feature segments are defined as follows.

A feature segment is a number of contiguous strictly positively or negatively attributing features over time within one dimension d of x that is at most 10 seconds long. This duration is assumed to capture the temporal dependencies at an appropriate level of information coarseness from the explanation. In other words, this means that 10 seconds are assumed to capture the temporal relations in the explanation. At the same time, the feature segment is short enough to provide a detailed basis for the evaluation of different attribution values, meaning that attribution values are not expected to vary strongly within 10 seconds if all attribution values are either positive or negative.

Feature segments are determined through the explanation obtained from TSA. Within an input dimension d (i.e., a sensor), the attribution values are inspected and if applicable, segments are formed with a maximum window length. The first segment is built around the highest attribution value in d at the middle of the window, and the following segments are defined accordingly. This approach of feature segments is followed in the evaluation of faithfulness and attribution sufficiency.

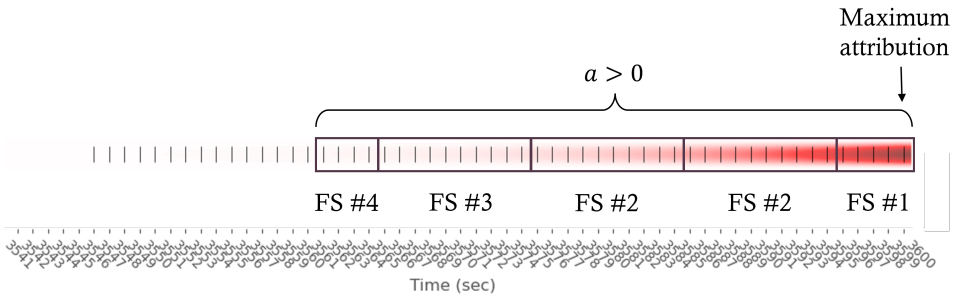


Figure 6.3: Feature segment (FS) definition based on attribution of input dimension d. Only non-zero attribution is considered. Segments with a maximum window size of 10 time steps are defined around the first segment which contains the maximum attribution.

Faithfulness

Faithfulness [42], also referred to as fidelity or truthfulness [36] is the first evaluation criterion of the technical evaluation. This refers to whether the explanation reflects the true behaviour of the model. In other words, faithfulness is a desirable quality that indicates whether the explainability achieved through the explanation method is faithful to the true reasoning of the model. I.e., the attribution values defined through the explanation reflect the SNN model and how it arrived at the prediction. This quality is universally desirable and holds the essence of XAI, which aims at finding methods to explain black-box models. Faithfulness is qualitatively evaluated in Jeyasothy et al. (2019)'s work [37] as reliability, as well as Kim and Panda (2021)'s work [40] as accuracy. While the latter used explanations from Grad-CAM as ground truth to obtain their evaluation, a different approach is taken in this thesis.

To measure faithfulness of TSA, the metric *explanation selectivity* [59] is chosen. This metric evaluates whether feature attributions defined by TSA are faithfully attributing respectively to the model's prediction. This is determined in an iterative process similar to Montavon, Samek, and Müller (2018) [59]. Input feature segments are ranked by their attribution and iteratively removed by their rank (i.e., highest attributing first). However, due to the nature of time series data, the removal of a feature segment is not straightforward. A feature is defined by its input dimension as well as time step, therefore removing it would interrupt the time series. Instead, the inversion of the feature values acts as the removal of features, similar to the perturbations proposed by Schlegel et. al (2019) [60] for the evaluation of XAI methods on time series data. The idea behind this is that changing the value of highly attributing features should lead to a

change in the model's prediction.

In detail, *explanation selectivity* [59] is defined as the area under the curve (AUC) of the graph resulting from inverting the most attributing feature segments first. A low explanation selectivity is desirable, as the performance is expected to drop significantly for highly attributing feature segments. Algorithm 2 describes the computation of this metric.

Algorithm 2 Explanation selectivity [59]

Let e be the explanation function that results in feature attribution map $A(x,t)$ describing the attributions to the predicted class, $f(x,t)$ denote the model's prediction on an input $x \in X$ at time t. Let R be the total number of feature segments of x, and N the size of the test set X and Y be the corresponding ground-truth labels for X.
for $x \in X$ **do**
  for $t = 1, 2, ..., T$ with T being the duration of x **do**
    $A(x,t) \leftarrow e(f,x,t)$
    Define R feature segments.
    Sort the feature segments in descending order by their mean attribution values.
    for rank $r = 0, 1, ..., R$ **do**
      $x^{\text{inv@}r} \leftarrow$Invert the value of feature segment $x^{(r)}$ so that $x^{(r)} = |x^{(r)} - 1|$.
      $\hat{y}^{\text{inv@}r} \leftarrow f(x^{\text{inv@}r}, t)$
    end for
  end for
end for
Let $X^{\text{inv@}r}$ denote X with feature segments up to rank r inverted.
for rank $r = 0, 1, ..., R$ **do**
  Compute Balanced Accuracy of $\hat{Y}^{\text{inv@}r}, Y$.
end for
Compute explanation selectivity as the AUC of the graph resulting from the performance of the model depending on the amount of feature segments inverted.

The experimental setup for the evaluation of faithfulness looks at the explanation selectivity of the TSA explanations of the predicted class.

Attribution Sufficiency

While faithfulness is a quality that answers the question of whether the feature attributions assigned by the explanation are true, what is referred to as attribution sufficiency in this thesis highlights another quality aspect of the explanation: the sufficiency of the set of important features in the sense of propositional logic for the prediction. This refers essentially to the question of whether the set of important features F is sufficient for the same model prediction $\hat{y}$, i.e., $F \rightarrow \hat{y}$ and is also referred to as fidelity or faithfulness in related work [61]. It addresses the input-output mapping of the explanation when compared to the original model, and is an interesting aspect for feature attribution based explanations. A feature-based explanation that is sufficient covers all important features that are relevant to the prediction. Attribution sufficiency is therefore strongly connected to faithfulness and is a desirable quality.

If the attributions defined by TSA are sufficient for the model prediction, the explanation consists of all relevant features to the same prediction as for the whole data by the model f. Therefore, the measurement of this quality looks at f's behaviour when presented with solely the important features. Similarly as is the case for the evaluation of faithfulness, eliminating features from time series data is not trivial due to the i.i.d. nature of the data. Unlike the inversion approach proposed for faithfulness, the elimination of features is handled by a random shuffling of the

unimportant features of the data. This is because faithfulness is evaluated through an iterative process, while the unimportant features are eliminated at once for the evaluation of attribution sufficiency. If all feature values were inverted, the distribution of the data would change, thus limiting the applicability of f. The data shuffling, however, does not change the distribution[62].

Therefore, *the degree of attribution sufficiency* describes the accuracy of the model f's performance using only the important features and shuffling the unimportant features with regards to its predictions $\hat{y}$ using all features. Unimportant features are defined as such, which do not belong to a highly attributing feature segment. In other words, unimportant features have an absolute attribution value defined by the explanation lower than a certain threshold and do not belong to a highly attributing feature segment. The absolute attribution value is taken as an indicator of importance as large attribution values regardless of their sign represent important information to the model prediction. High sufficiency is desirable, as model performance is expected to be similar for the clean and perturbed input.

Algorithm 3 Degree of sufficiency

Let $f(x,t)$ be a SNN model's prediction on input $x \in X$ at time t, e be the explanation function which results in attribution map $A(x,t)$ that describes the attribution to the predicted class. Let ϵ be the threshold for feature importance.
for $x \in X$ **do**
  for $t = 1, 2, ..., T$ with T being the duration of x **do**
    $A(x,t) \leftarrow e(f,x,t)$.
    Mask A where $|a| > \epsilon$.
    $x_p \leftarrow$ Perturb unmasked area of A.
    $\hat{y}_p \leftarrow f(x_p, t)$
  end for
end for
Compute the degree of sufficiency as the balanced accuracy of $\hat{Y}_p, \hat{Y}$.

Similar to faithfulness, the attribution sufficiency is determined by the TSA explanations for the predicted class. As an ideal ϵ for identifying the important features is not known beforehand, eight different values of ϵ are tested. Namely, these are 0, 25%, 50% and 75% of the maximum absolute feature attribution recorded over all extracted explanations. Additionally, the thresholds 5%, 10%, 15% and 20% were added to the experimental set-up after receiving initial results with the purpose to inspect TSA's attribution sufficiency for lower values of ϵ. The thresholds are defined relative to the maximum attribution due to the unnormalised nature of the feature attributions defined through TSA.

Sensitivity

Next to faithfulness and attribution sufficiency, an explanation's trustworthiness is dependent on its stability and robustness [36, 48]. These are qualities that refer to an explanation's behaviour when small perturbations are added to the input, which is referred to as an explanation's *sensitivity* [63] in this report. An explanation is stable if it does not change strongly for similar input caused by natural perturbations in the data due to e.g. noise. Robustness is achieved if an explanation does not change strongly to intended perturbations, i.e. adversarial attacks [48]. However, the evaluation of robustness is not in the scope of this thesis as it first raises different research questions in this area: As adversarial examples intentionally mislead the model, can robustness only be measured of adversarially robust models? How are adversarially robust SNNs built? Should a feature-based explanation method such as TSA highlight the features that lead the model to misclassify? Even though first studies have shown the increased adver-

sarial robustness of SNNs when compared to ANNs [16], this requires further research and is therefore out of the scope of this thesis. Nevertheless, both criteria indicate the sensitivity of the explanation to perturbed input, where an explanation with high sensitivity responds to perturbed input and explanations with rather low sensitivity do not change much. Low sensitivity is therefore desirable as high sensitivity is difficult for a user to understand, which could lead to distrust in the explanation. Furthermore, explanations with low sensitivity have a certain level of generalisation and are considered simpler than highly sensitive explanations [63].

A specific case of sensitivity is the behaviour of the explanation for the same input. In this case, the same explanation should be generated for all exact inputs x. It is a desirable quality since it is strongly connected to the reliability of an explanation method. However, this specific case is not measured by a metric. Instead, it is a binary property that an explanation method either fulfils or not. The explanation method presented in this report is generated without randomness as a post-hoc explanation with the inference model, whose model weights are fixed. The computation consists mainly of matrix multiplications, which are deterministic. Thus, a consistent explanation behaviour for the same inputs is ensured and it is not tested in the experiments for the evaluation of the explanation method.

Beyond the specific case of the same inputs, sensitivity as an explanation quality can be measured using *max-sensitivity* (6.1) as proposed by Yeh et al. (2019) [63]. Max-sensitivity is a metric that measures the maximum change in an explanation $e(f, x, t)$ given perturbations in a neighbourhood r around the input x at time t. The neighbourhood ensures the similarity of the original input to the perturbed input. This can be applied for both natural, unintended perturbations as well as adversarial attacks to measure stability and robustness respectively.

$$\text{Max-Sensitivity}(e, f, x, t, r) = \max_{||x'-x|| \leq r} ||e(f, x', t) - e(f, x, t)|| \tag{6.1}$$

Natural and unintended perturbations have to be simulated to test the stability of the explanation method. In the frame of the ADL dataset, natural perturbations cannot be modelled as the addition of random noise for example, since this would lead to new examples that are out of distribution and therefore also not similar to the original input. Instead, natural perturbations in an activity dataset with binary sensors could be modelled through a change in the sensor activation duration. This could occur in reality as different people usually follow a different routine, e.g., taking less or more time to shower. Moreover, also within data collected from one user, such natural perturbations could occur within different days.

To implement these perturbations, the duration of the active sensors within input x is randomly changed within the realms of 10% of its original duration. It is randomly chosen whether the activation is lengthened or shortened if it is perturbed at the start of the activation or the end, as well as by how much exactly. By limiting the perturbation to 10% of the original duration, the similarity between the perturbed and original data shall be ensured. Each of the samples in the test set for technical evaluation is perturbed this way and explanations are extracted using TSA. Then, *max-sensitivity* is computed as the maximum Frobenius norm of the difference between the explanations on clean and perturbed data. A small difference is to be expected, as the data is different. However, the value should be small if TSA is stable.

As a baseline for the evaluation of explanation sensitivity, random baseline explanations based on the extracted explanations from the perturbed input are generated. This is done similarly to the generation process of the baseline explanations for the evaluation of the explanations on original data. I.e., the features with non-zero attributions of the TSA explanations are randomly assigned an attribution value within the interval of the minimum and maximum recorded attribution. To compute the baseline stability, the max-sensitivity of the baseline explanations on the clean data is compared to the baseline explanations of the perturbed data.

6.1.2 Results and Discussion

The technical experiments measuring faithfulness, attribution sufficiency and stability are conducted through automated Python scripts[12] run on the high performance computing cluster at the Institute for Artificial Intelligence in Medicine of the University of Duisburg-Essen. The results are presented and discussed in the following.

Faithfulness

The results of the evaluation of faithfulness are presented in Table 6.1 and the graphs resulting from the iterative feature segment removal is shown in Figure 6.4. The explanations extracted from the different models have a different maximum number of feature segments detected in the data, which is why there is variation in the y-axis scale.

Figure 6.4 shows two types of graphs per SNN model: firstly, the model performance with regards to the ground truth (i.e., balanced accuracy of $Y, \hat{Y}_p$) in the bottom row, and secondly, the model performance with regards to the original model predictions (i.e., balanced accuracy of $\hat{Y}, \hat{Y}_p$) in the top row. The latter is the basis for the explanation selectivity score because it reflects the feature segment's attribution to the prediction, whereas the first shows the decline of model performance to chance as the number of flipped segments increases. Both curves are not necessarily similar as e.g. the curves of *OneLayerSNN* show from roughly the 100$^{\text{th}}$ segment on. In these cases, original model predictions, which were wrong, may be predicted correctly with the perturbed input.

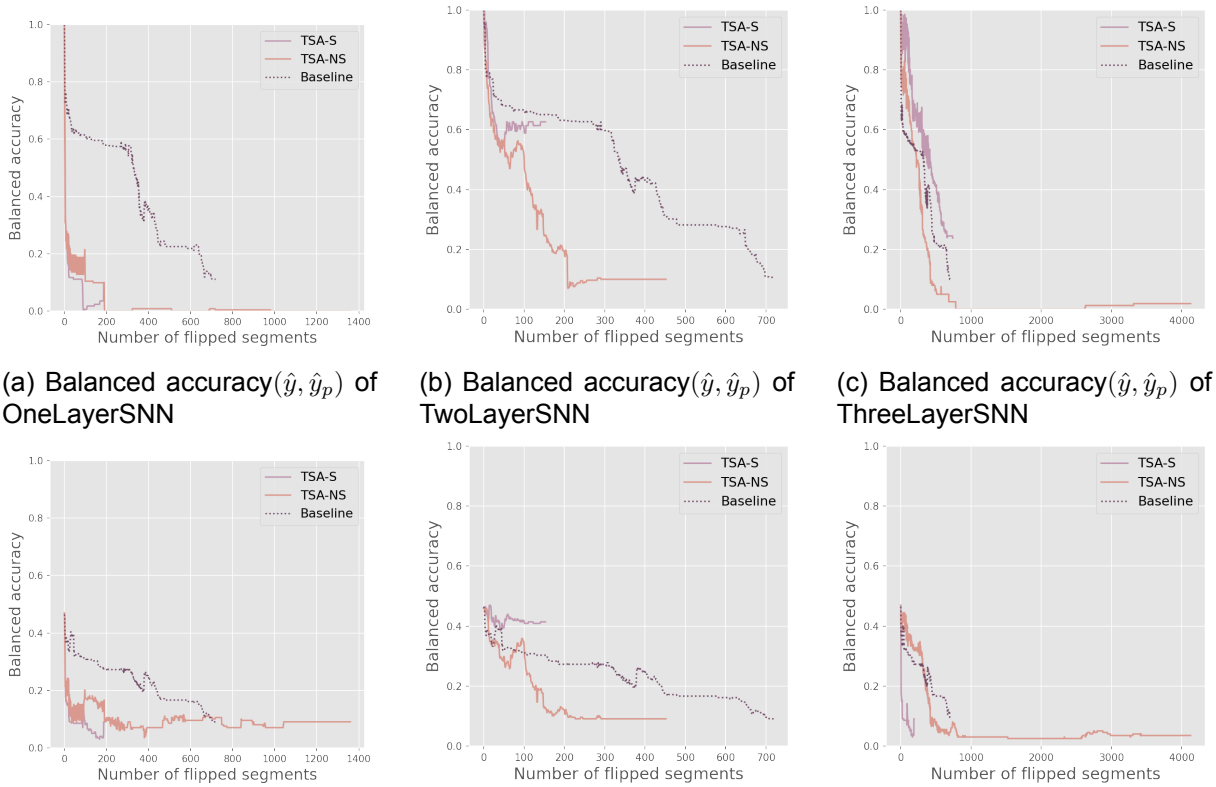
Model	**TSA-S**	**TSA-NS**	**Baseline**
OneLayerSNN	0.086 ± 0.041	$\mathbf{0.024 \pm 0.022}$	0.411 ± 0.072
TwoLayerSNN	0.635 ± 0.070	$\mathbf{0.248 \pm 0.063}$	0.462 ± 0.073
ThreeLayerSNN	0.541 ± 0.073	$\mathbf{0.061 \pm 0.035}$	0.392 ± 0.071
Average	0.421 ± 0.072	$\mathbf{0.111 \pm 0.046}$	0.422 ± 0.072

Table 6.1: Explanation selectivity of the TSA explanations extracted per model and on average reported at 95% CI. It is measured as the AUC of the plots (a), (b), (c) of Figure 6.4.

The explanation selectivity of the TSA-NS explanations is low compared to both TSA-S explanations and the random baseline. This implies that TSA-NS generates faithful explanations whereas TSA-S explanations are less faithful to model behaviour. Therefore, the absence of spikes in the data is indicated to be relevant to the model prediction. As TSA-S does not consider these attributions, it is not completely faithful to model behaviour. TSA-S explanations achieve a similar overall explanation selectivity to the baseline, at first glance implying that the TSA-S explanations are not better than random explanations in faithfulness to model behaviour.

Upon further inspection, the explanation selectivity scores per model suggest that TSA-S does provide faithful explanations and significantly outperforms the baseline in explaining *OneLayerSNN* (0.086 ± 0.041 vs. 0.411 ± 0.072). For the other models, the baseline explanations are better than the explanations provided by TSA-S. This observation indicates that TSA-S is not a faithful explanation method for deep SNNs. The propagation of the attribution scores across the different layers might not capture the correct model behaviour in the explanation. Nonetheless, the number of segments in the baseline explanations and TSA-S explanations must be considered when comparing them. The baseline explanations are larger and therefore exhibit more

[12] The scripts for the technical evaluation are available at: https://github.com/ElisaNguyen/tsa-explanations.



(a) Balanced accuracy$(\hat{y}, \hat{y}_p)$ of OneLayerSNN

(b) Balanced accuracy$(\hat{y}, \hat{y}_p)$ of TwoLayerSNN

(c) Balanced accuracy$(\hat{y}, \hat{y}_p)$ of ThreeLayerSNN

(d) Balanced accuracy$(y, \hat{y}_p)$ of OneLayerSNN

(e) Balanced accuracy$(y, \hat{y}_p)$ of TwoLayerSNN

(f) Balanced accuracy$(y, \hat{y}_p)$ of ThreeLayerSNN

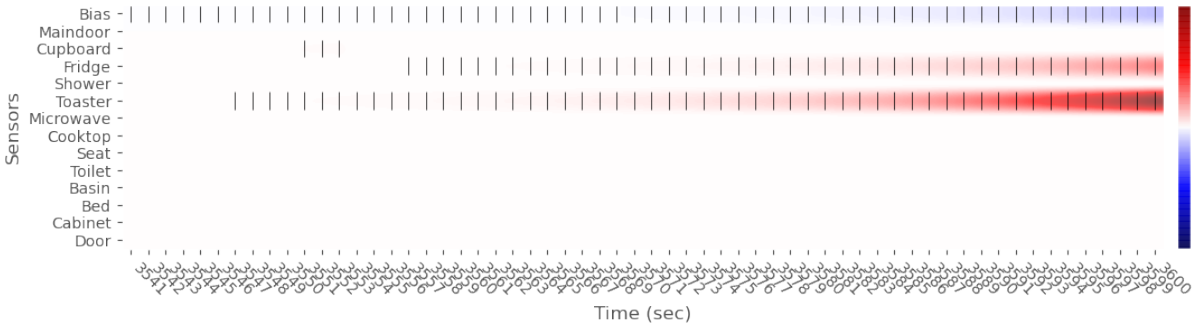
Figure 6.4: Faithfulness of TSA variants as balanced accuracy per model with flipped feature segments on the test set for technical evaluation with the respective baselines.

feature segments, whereas the number of feature segments for TSA-S explanations is comparatively low. Hence, the explanation selectivity score does not capture the fact that the drop in accuracy within flipping the first few feature segments is stronger with TSA-S than with the baseline explanations for *OneLayerSNN* and *TwoLayerSNN* (see Figures 6.4a and 6.4b). Thus, the definition of the baseline explanations may not be ideal for the comparison because TSA-S does seem to faithfully capture parts of the true model behaviour in the first feature segments.

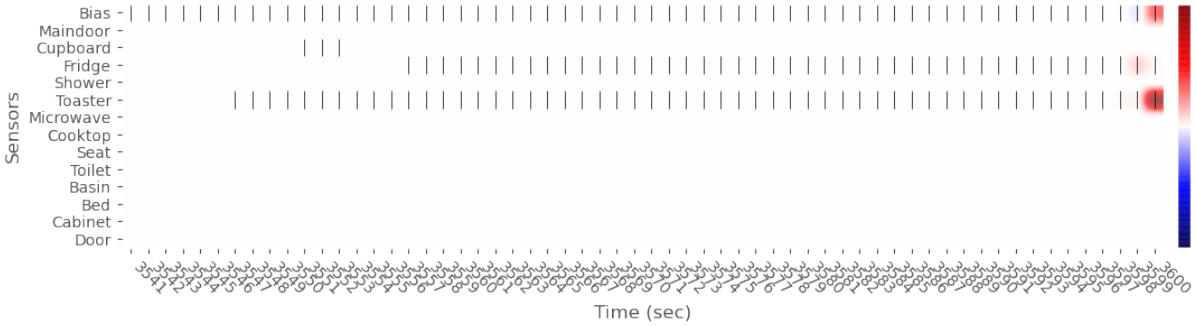
Generally, the graphs resulting from iterative flipping of the ranked feature segments all fulfil the expectation of a sharp decrease at first followed by a slower decrease for both TSA variants. As can be seen in Figure 6.4, the performance decreases sharply with the flipping of the highest-ranking feature segments for both TSA explanations extracted from all models. In contrast, the baseline explanations stagnate in the drop in performance when reaching a balanced accuracy of around 0.6. This observation suggests that the high ranking feature segments carry the highest importance for the model prediction, and TSA, specifically TSA-NS, is rather successful in faithfully identifying these segments.

It is noticeable that the number of feature segments is larger for the explanations extracted from *OneLayerSNN* and *ThreeLayerSNN*. These models have a slower decay of their neuron's membrane potential. Consequently, the TSA explanations are larger because the attribution of past timesteps decays slower than for *TwoLayerSNN* (Figure 6.5). Hence, the explanation consists of more non-zero attribution values than the explanations from *TwoLayerSNN*, which consequently leads to a higher number of feature segments. This explains the different y-axis scales of the different models. Within the models, TSA-S explanations are much smaller than TSA-NS explanations judging from the number of feature segments, therefore the performance

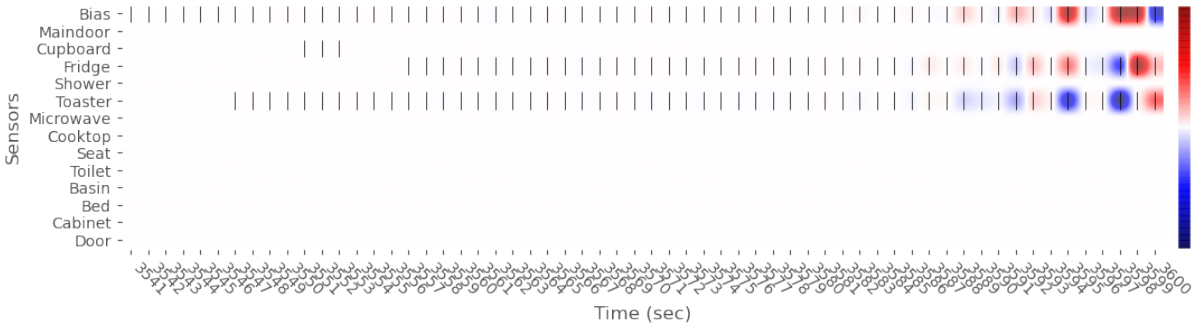
graphs stop at a certain number of feature segments in Figure 6.4.



(a) TSA-S explanation extracted from *OneLayerSNN* for timestep 177081 of the testset.



(b) TSA-S explanation extracted from *TwoLayerSNN* for timestep 177081 of the testset.



(c) TSA-S explanation extracted from *ThreeLayerSNN* for timestep 177081 of the testset.

Figure 6.5: Example explanations to visualise the effect of the membrane potential decay rate on the attribution. *OneLayerSNN* and *ThreeLayerSNN* have a slower decay rate, which shows in the larger non-zero attribution values further in the past, indicated by the intensity of the attribution colours. Hence, the number of feature segments for *TwoLayerSNN* is significantly smaller.

Upon inspection of the faithfulness per SNN model, it is evident that the TSA explanations are the least faithful to *TwoLayerSNN*. The explanation selectivity score is highest for both TSA variants as well as the baseline for this model. This observation could be linked to the membrane potential decay rate, as this is a property that sets *TwoLayerSNN* apart from the other models. It is also noteworthy that the balanced accuracy recovers to roughly 0.6 after around 50 flipped feature segments identified with TSA-S. This is a strong indication that TSA-S does not faithfully explain the model prediction for *TwoLayerSNN*. The explanation seems to be incomplete because a decay of model performance to chance is expected, whereas the model performance only decreases by 0.1 to 0.4 (Figure 6.4e). For TSA-NS, the balanced accuracy also shortly increases again at around feature segment 50 before continuing to definitely decrease at roughly 100 feature segments flipped. This fluctuation could be reasoned by the correct classification

of some samples by chance as the balanced accuracy continues to decrease steadily again. The lowest explanation selectivity scores are achieved for TSA explanations of *OneLayerSNN*'s predictions, which implies that TSA explanations for this model are most faithful. TSA in both variants seems to capture the true model behaviour of *OneLayerSNN* specifically well. It is interesting to note that the balanced accuracy of *ThreeLayerSNN*'s predictions with flipped feature segments with regard to the original model predictions decreases the fastest for the baseline explanations (Figure 6.4c). While the drop in performance for feature segments identified from TSA-S and TSA-NS explanations is also steady and sharp, the random baseline explanations lead to the fastest decrease within the first 100 feature segments. For *OneLayerSNN*, the opposite is clearly the case: The performance of the model predictions with flipped feature segments identified from the TSA explanations clearly decreases stronger and faster than the baseline. Therefore, there is an indication that TSA as an explanation method loses faithfulness with the addition of layers in the SNN model to be explained.

Overall, the evaluation of faithfulness using the explanation selectivity score [59] yielded expected results which indicate the faithfulness of TSA explanations. The inversion of the highest attributing feature segments leads the model to classify differently than with the original data in all models and with both TSA variants. The general faithfulness of TSA, especially TSA-NS, is shown in this experiment, which validates the approach of TSA as an explanation method.

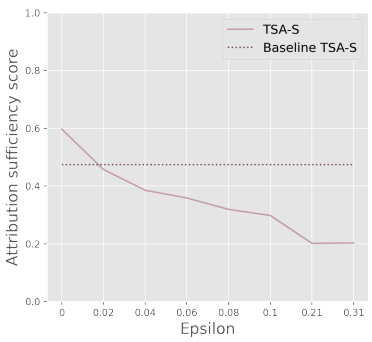
Attribution Sufficiency

The degrees of attribution sufficiency given different thresholds for the consideration of highly attributing feature segments (defined through ϵ) are shown in Figure 6.6. The concrete value of ϵ varies between the TSA variants because the explanations extracted with TSA-S and TSA-NS have different maximum attribution values which the definition of ϵ is based on. The overall attribution sufficiency is reported as the mean of the computed sufficiency scores across the different underlying SNN models. As can be seen in Figure 6.6, the sufficiency of the explanations is only larger than of the baseline explanations for $\epsilon = 0$. Therefore, the overall attribution sufficiency scores are reported at a 95% CI in Table 6.2 for $\epsilon = 0$.

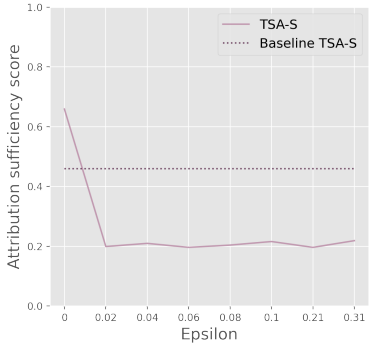
Model	**TSA-S**	**TSA-NS**	**Baseline**
OneLayerSNN	0.597 $\pm$ 0.072	**1.000 $\pm$ 0.000**	0.475 $\pm$ 0.073
TwoLayerSNN	0.659 $\pm$ 0.069	**0.926 $\pm$ 0.038**	0.460 $\pm$ 0.073
ThreeLayerSNN	0.546$\pm$ 0.073	**0.960 $\pm$ 0.028**	0.445 $\pm$ 0.073
Average	0.601 $\pm$ 0.072	**0.961 $\pm$ 0.028**	0.460$\pm$ 0.073

Table 6.2: Attribution sufficiency scores of the TSA explanation method at 95% CI for $\epsilon = 0$.

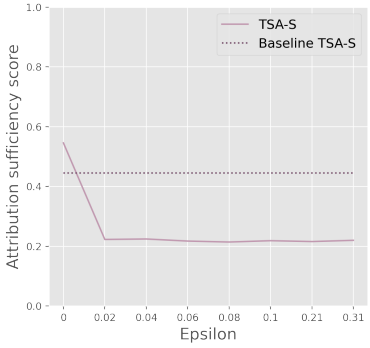
Figure 6.6 clearly shows that the best value for ϵ is 0 for the explanations extracted from all models for both TSA variants. For this threshold, the attribution sufficiency of TSA-NS is particularly large. The TSA-S explanations also show improved sufficiency performance in comparison to the baseline at $\epsilon = 0$, but not as large as the sufficiency scores of TSA-NS. A possible reason for this is the definition of the background close to the time of prediction in the case of TSA-S. The difference in performance between TSA-S and TSA-NS supports this observation. The TSA-S explanations consist of attribution scores of spiking feature segments exclusively (Figure 6.7a), while TSA-NS also assigns attributions to non-spikes in the data (Figure 6.7c). Therefore, it is less likely for TSA-NS to consider recent data of the time series (i.e., in the timeframe shortly before the model prediction) as background. Hence, it is less likely that this area is perturbed (Figure 6.7). Nevertheless, the evaluation is still valid as unimportant features should not influence the model prediction strongly, regardless of their value. This observation indicates that



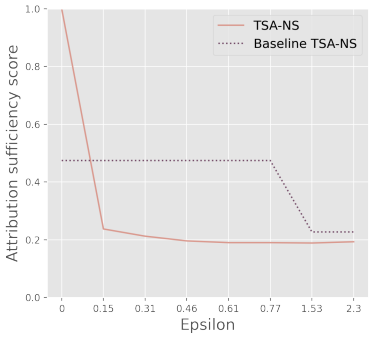
(a) Sufficiency of TSA-S explanations extracted from OneLayerSNN



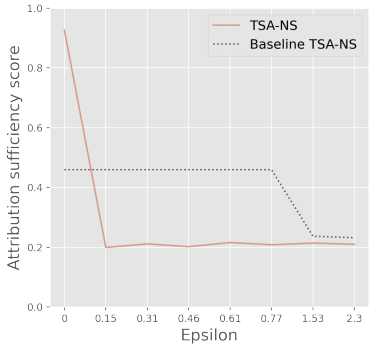
(b) Sufficiency of TSA-S explanations extracted from TwoLayerSNN



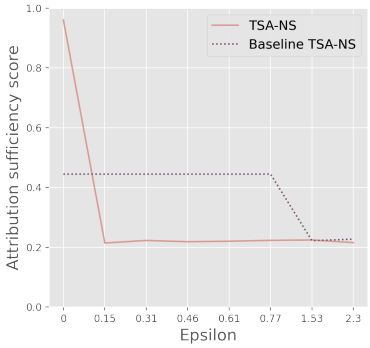
(c) Sufficiency of TSA-S explanations extracted from ThreeLayerSNN



(d) Sufficiency of TSA-NS explanations extracted from OneLayerSNN



(e) Sufficiency of TSA-NS explanations extracted from TwoLayerSNN



(f) Sufficiency of TSA-NS explanations extracted from ThreeLayerSNN

Figure 6.6: Attribution sufficiency from TSA explanations per model with their respective baselines per ϵ values 0, 5%, 10%, 15%, 20%, 25%, 50%, 75% of the extracted maximum attributions.

non-spikes are relevant to the prediction, as already implied by the faithfulness experiments.

Taking a threshold $\epsilon > 0$ does not capture a set of feature segments that are sufficient for the original model predictions because the model requires all attributing segments for its prediction. Instead, every feature segment that exhibits an absolute attribution value larger than 0 belongs to the required set of feature segments for a sufficient explanation. This means that there is no large difference between the attributing feature segments in their importance with regard to sufficiency. The attribution value is secondary as long as it is non-zero for sufficient explanations. Therefore, the TSA explanations are evaluated with $\epsilon = 0$.

The average degree of sufficiency of 0.961 ± 0.028 indicates a high degree of sufficiency for TSA-NS explanations. In comparison to the degree of sufficiency of both TSA-S and the baseline explanations, TSA is significantly superior in this evaluation. The explanations extracted from *OneLayerSNN* even display a perfect attribution sufficiency of 1, showing that explanations of this model are sufficient to the original model prediction. This high score could be caused by the slower decay of the membrane potential compared to *TwoLayerSNN*. As the neuronal contribution score (NCS), which represents the model spikes component in the computation of TSA, depends on the decay rate of the model it is explaining, the attribution values farther in the past more likely carry a non-zero value. This leads to parts of the input farther in the past being marked as highly attributing (i.e., $|a| > \epsilon$) so that the background is smaller. Thus, the perturbation applied to the background of the samples in the frame of the attribution sufficiency experiment would have little effect on the last time steps before the prediction. Hence, the pre-

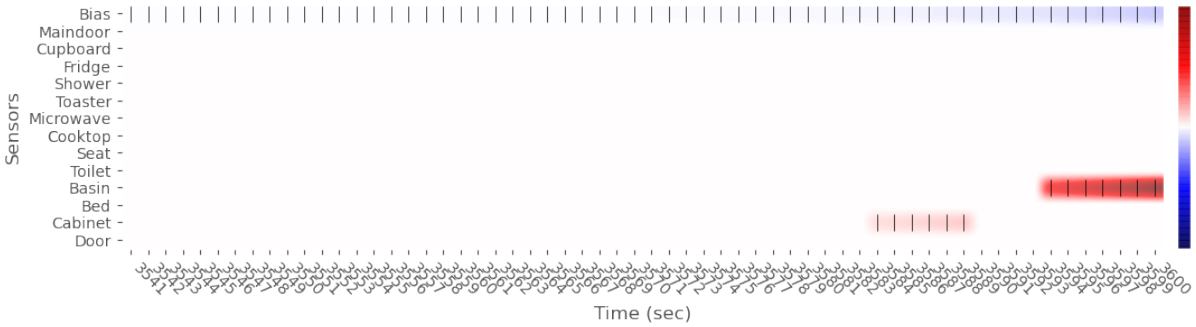
diction is not strongly influenced by the background perturbation, enabling a perfect sufficiency score for *OneLayerSNN*. An example of this can be seen in Figure 6.7 and examples from the other models can be found in appendix C.

ThreeLayerSNN has the same decay rate of the membrane potential as *OneLayerSNN*. As the degree of sufficiency for the TSA-NS explanations extracted from *ThreeLayerSNN* is larger than for the TSA-NS explanations from *TwoLayerSNN*, the role of the decay rate in connection with the background size with regard to attribution sufficiency is supported. For the TSA-S explanations, however, the explanations extracted from *TwoLayerSNN* achieve the highest degree of sufficiency. In this case, the decay rate is likely responsible for this result but differently than for TSA-NS explanations. The model relies on recent parts of the data in a small time interval. Random shuffling perturbations of the background are less probable to perturb this small time interval. Therefore, the highly attributing feature segments of the TSA-S explanations from *TwoLayerSNN* achieve a higher degree of sufficiency.

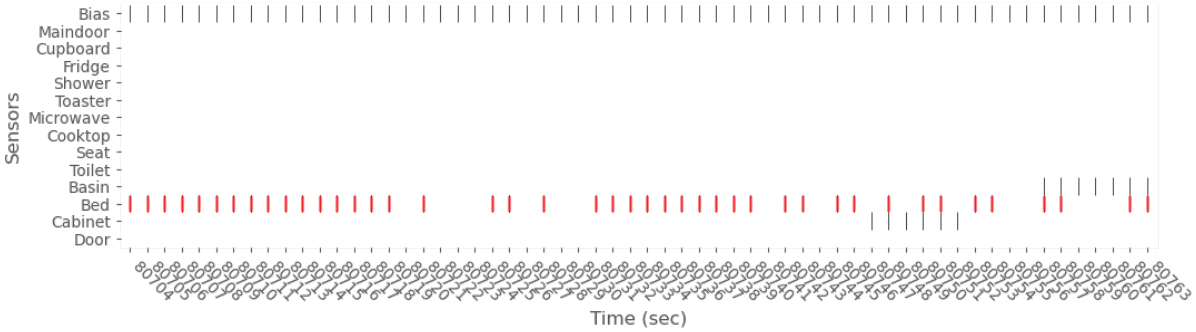
However, TSA-NS explanations from *ThreeLayerSNN* do not achieve perfect attribution sufficiency either. The reason for this difference could lie in the more complex input-output mapping in deeper models. In *OneLayerSNN*, each input dimension is directly connected to the output neurons, leading to rather direct effects showing in the perturbation of the input. TSA is based on a forward pass approximation of the model it is explaining, meaning that these direct changes are more immediate in the extracted explanation. In contrast, deeper networks propagate the input across hidden computational layers, which is considered in the computation of TSA. Nevertheless, each attribution value uses all spike and weight components of the hidden layers. This effect of hidden layer computation influences the computation of the explanation, evidently leading to slightly less sufficient explanations using both TSA-S and TSA-NS in deeper models. The respective baseline sufficiency scores per model support this, as they also decrease with an increase of model layers.

At the same time, the depth of the model is connected to the decay of feature attribution values, too. As the weighted NCS' of each layer are multiplied with the weighted NCS' of the following layers, the attribution score consists of as many multiplications of values ≤ 1 as there are layers. With a larger number of layers, this causes the attribution score to be small. Therefore, the explanations extracted from deeper models such as *ThreeLayerSNN* assign non-zero attribution values to features closer to the current timestep when compared to shallow models like *OneLayerSNN*. By the same logic as the argument of the decay rate, this consequently leads to larger parts of the data to be considered as background.

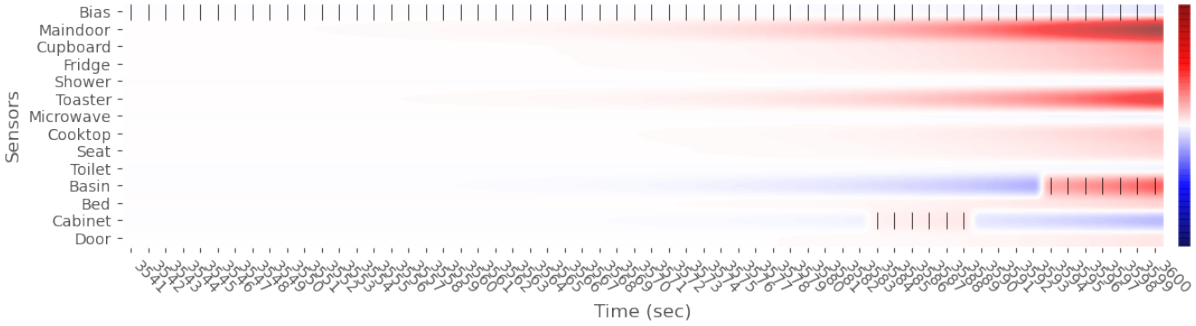
Thus, the evaluation of attribution sufficiency shows that explanations extracted from SNNs using TSA-NS show a high degree of sufficiency in the feature attributions, given the threshold for considering absolute attributions as important lies at $\epsilon = 0$. The explanations extracted with TSA-S score lower due to the exclusion of non-spikes as attributing features but still demonstrate a clear improvement from the baseline.



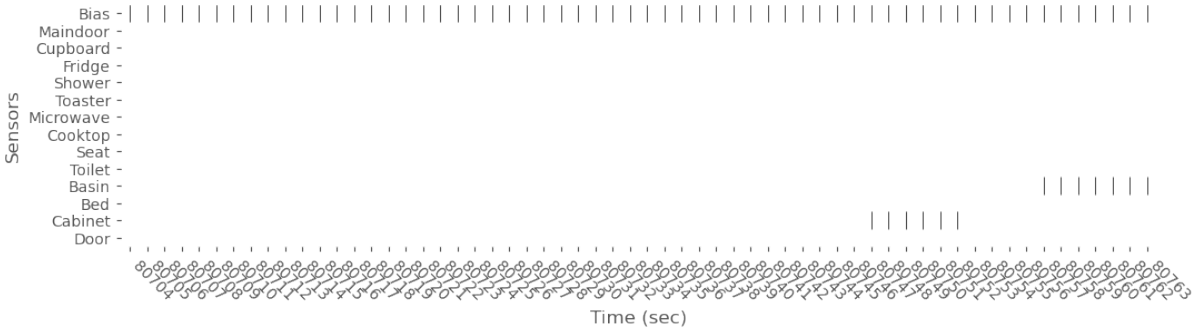
(a) TSA-S explanation for *OneLayerSNN*'s prediction of timestep 80763.



(b) Time series until timestep 80763 with perturbed background based on the TSA-S explanation of *OneLayerSNN*'s prediction.



(c) TSA-NS explanation for *OneLayerSNN*'s prediction of timestep 80763.



(d) Time series until timestep 80763 with perturbed background based on the TSA-S explanation of *OneLayerSNN*'s prediction but the background perturbation is not visible in the last 60 seconds.

Figure 6.7: Example of TSA-S and TSA-NS explanations of *OneLayerSNN*'s prediction for timestep 80763 in (a) and (c). Red marks positive and blue negative attributions. (b) and (c) show the data with a shuffled background based on the explanations and $\epsilon = 0$. The red spikes show the difference between the perturbed and clean input. While the bed sensor activation from earlier in the time series is shuffled in the last 60 seconds in the case of TSA-S, TSA-NS considers all sensor dimensions as relevant, hence the last 60 seconds are not perturbed.

Sensitivity

Table 6.3 presents the max-sensitivity scores per model for TSA-S and TSA-NS explanations compared to the random baseline explanations. As the baseline is random and does not depend on a model, only one baseline score is reported. In all cases, the TSA explanations outperform the baseline strongly, with the explanations extracted with TSA-S from *TwoLayerSNN* achieving the best score of 0.033. Also for TSA-NS explanations, the max-sensitivity for *TwoLayerSNN* is lowest. Figure 6.8 displays an example of the TSA explanations extracted from the predictions of *TwoLayerSNN* on the clean and perturbed input. Examples of explanations for the other models can be found in appendix C.

Model	TSA-S	TSA-NS	Baseline
OneLayerSNN	**0.853**	2.359	-
TwoLayerSNN	**0.033**	0.450	-
ThreeLayerSNN	**0.316**	6.733	-
Average	**0.401**	3.181	175.923

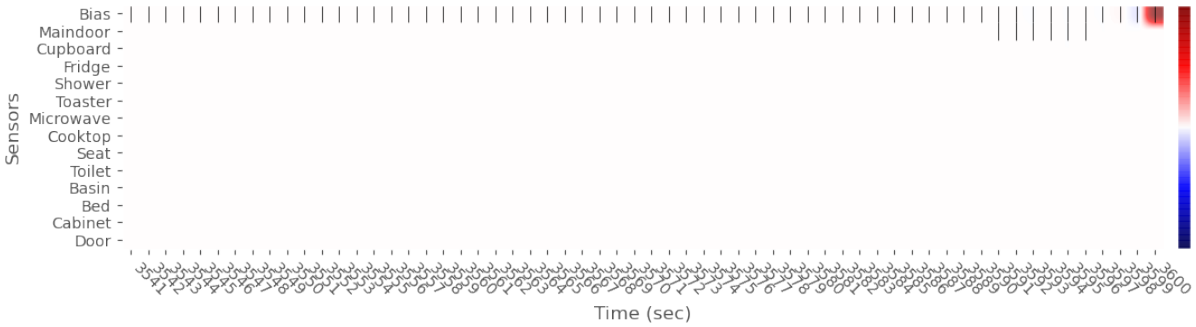
Table 6.3: Max-sensitivity results of the TSA-S and TSA-NS explanations extracted per model and on average with their respective baselines.

The results clearly show that the TSA explanations provide stability concerning the tested perturbation of random shortening and lengthening of spike trains, where TSA-S is more stable than TSA-NS. Compared to the random baseline, the average max-sensitivity of both TSA-S (0.401) and TSA-NS (3.181) is undoubtedly smaller. This indicates a high level of stability of TSA explanations with regard to such perturbations. However, a baseline of random attributions may be rather unfit for this type of experiment, as it sets the baseline up for particularly poor performance. Nevertheless, the results demonstrate that the TSA explanations are not random, and perform significantly better than random initialisation of feature attributions.

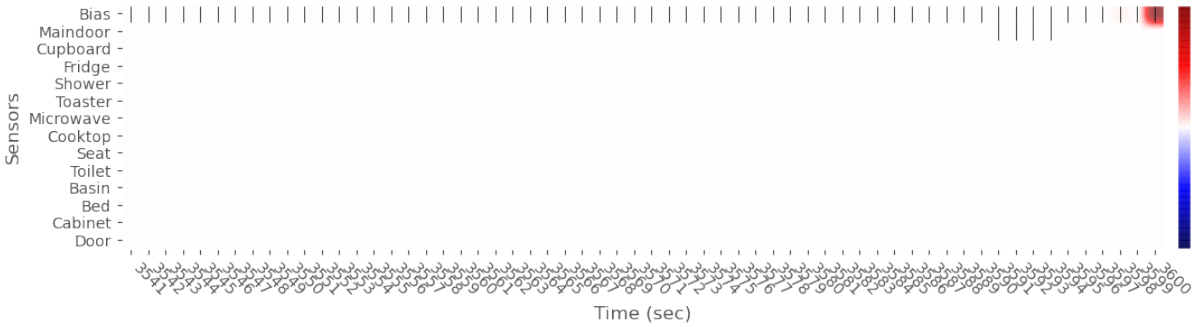
TSA-S outperforms TSA-NS noticeably in terms of stability. The max-sensitivity scores of the explanations extracted per model are in each case lower for TSA-S explanations. Hence, TSA-S explanations are less sensitive to natural perturbations than TSA-NS explanations. This can in part be explained by the nature of considering non-spiking attribution as 0 like in TSA-S explanations. The non-spiking part of the input cannot exhibit a non-zero attribution value, which is why there is a smaller number of attributing features in TSA-S explanations as opposed to TSA-NS (e.g. Figure 6.8a vs Figure 6.8c). Therefore, changes in the attribution values are likely less prominent, leading to a lower Frobenius norm of the attribution map matrix. Consequently, the max-sensitivity score is likely small.

The explanations extracted from *TwoLayerSNN* produce the lowest sensitivity to the tested types of natural perturbations for both TSA variants, therefore they exhibit the highest level of stability. In this case, the larger membrane potential decay rate of *TwoLayerSNN* when compared to the other networks could be a reason for the improved stability (Figure 6.8). Due to the high decay rate, the explanations extracted from this model are smaller in the sense that they provide a smaller number of features with attribution values. As the number of features with attribution is lower than in the explanations extracted from the other models, the Frobenius norm that defines the max-sensitivity score is also lower. The explanations from *TwoLayerSNN* happen to be smaller in terms of the Frobenius norm, still, the size of the attribution maps extracted from all models is the same, thus they are comparable.

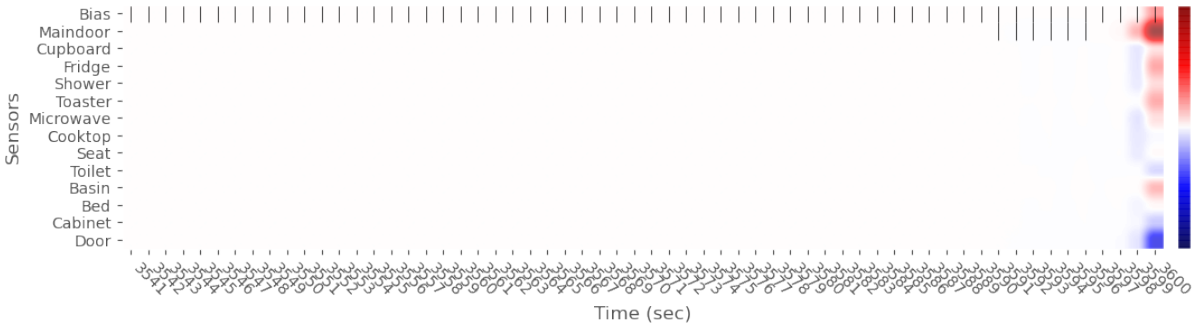
Interestingly, the explanations extracted from *ThreeLayerSNN* (Figure C.4) are more stable than from *OneLayerSNN* (Figure C.3) in TSA-S while it is the other way around for TSA-NS explanations. TSA-S displays the expected behaviour in this regard: Based on the direct mapping from



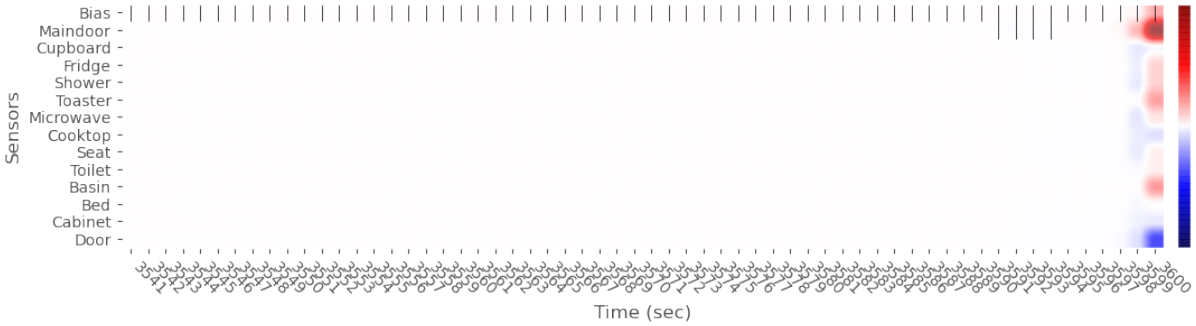
(a) TSA-S explanation for *TwoLayerSNN*'s prediction of timestep 26442.



(b) TSA-S explanation for *TwoLayerSNN*'s prediction of timestep 26442 with natural perturbation.



(c) TSA-NS explanation for *TwoLayerSNN*'s prediction of timestep 26442.



(d) TSA-NS explanation for *TwoLayerSNN*'s prediction of timestep 26442 with natural perturbation.

Figure 6.8: Example of TSA-S and TSA-NS explanations of *TwoLayerSNN*'s prediction for timestep 26442 with and without natural perturbation (shortening of the maindoor activation by two seconds). Red marks positive and blue negative attributions. The explanation does not show large changes for the perturbed example in both TSA variants, most likely due to the decay rate of the neuron's membrane potential.

the input layer to the output layer, changes in the input were expected to show more strongly

in the explanations from *OneLayerSNN*, but this is not the case for TSA-NS explanations. Instead, the sensitivity of *ThreeLayerSNN* is almost three times as large (6.733 vs. 2.359). This indicates that the amount of layers, as well as their size, plays a role in the evaluation of explanation sensitivity, too. As the attribution scores for the explanation are computed using a summation over the weighted neuronal contribution scores (NCS), TSA for larger models could tend towards larger attribution values in the explanations. Thus, this evaluation highlights the need for some form of normalisation with regard to hidden layer size in the TSA calculation. Nonetheless, the evaluation of sensitivity infers a high level of stability for TSA explanations, with TSA-S demonstrating improved stability in comparison to TSA-NS. As robustness is not evaluated, a general statement about TSA's sensitivity cannot be made.

6.2 User evaluation

The objective of explanations and efforts in XAI is to provide interpretability for a model, which is the ability to be understood by humans. Therefore, the human-comprehensibility, especially by the target group of the explanation method, is a quality to measure an explanation by [36]. Comprehensibility is therefore a human-centred quality of explanations, which does not have a clear definition. While the time series specific evaluation frameworks presented in [42] and [48] do not mention human-centred metrics, the human-comprehensibility is a central part of XAI [6]. Hence, the explanation developed in this thesis is evaluated in this regard in a human-grounded evaluation [58]. Comprehensibility is a large concept and comprises other connected qualities.

A good and comprehensible explanation is clear in its intention, it is unambiguous [35]. This means that the explanation does not leave room for interpretation by the user; it shows clearly why a model made a certain decision. To measure *clarity*, a qualitative evaluation with users is required. The number of different user understandings acts as an indicator for clarity and ambiguity, where a clear explanation with little difference in understanding is desirable.

At the same time, a good explanation that is unambiguous to the user requires the explanation to match with the user's prior beliefs [36]. Explanations that do not fulfil this are at risk of devaluation due to confirmation bias, which states that people tend to ignore and devalue information that does not match their beliefs [64]. Measuring this stand-alone is difficult. As it is strongly connected to *clarity*, it is implicitly measured through the aforementioned process as well. Additionally, the user study is extended by a user study simulation that follows the forward simulation approach defined by Doshi-Velez and Kim (2017) [58], where the understanding of the explanation is tested. Users are asked to simulate the model behaviour based on the explanation. The idea behind this is that good explanations that are unambiguous and match the user's prior knowledge and beliefs should suffice to predict the model's classifications.

6.2.1 User Study Design

Especially for qualities like comprehensibility, it is clear that a purely technical, functionally-grounded evaluation is not satisfactory. A qualitative human-centred, and more specifically a user-centred evaluation must include the users themselves [6]. To evaluate qualitatively in a reproducible way, this study aims to provide a clear experimental setup.

To evaluate TSA explanations, the best performing model in terms of predictive accuracy is chosen as the model to be explained (*TwoLayerSNN*). Higher predictive accuracy implies that the model has learned the task better than the other models, and thus predictions are less based on random guessing. Additionally, this choice is made to limit the duration of the user study, as it increases the chance of participants. The focus is on evaluating TSA as an SNN model-

agnostic method so that explanations from the other models can be neglected in the evaluation of comprehensibility.

For the user study, six explanations for the predictions of *TwoLayerSNN* are extracted from the test set using TSA-S. These are selected to show a range of different types of data and model behaviour (i.e., misclassification and correct classification). To identify these samples, the test set is randomly sampled one by one, and the first six suitable samples are selected. These are examples of correct and incorrect classifications of data that differ from each other (e.g., not showing multiple samples with the *Bed* sensor activated even though this is quite common in the dataset).

The type of user study is a survey. Surveys are an efficient way to reach a large population and are appropriate for gaining insights into the experiences and attitudes of users [65]. As the aim of the study is to measure the comprehensibility of the explanation, which is an experience of the user, a survey is a fitting study type. The survey design method by Müller [65] is followed: (1) Research goal formulation, (2) Matching of research goals to constructs, and (3) Conversion of constructs to survey questions. The ethics approval for this user study can be found in the attachment.

The goal of the survey is the measurement of comprehensibility of the explanation method with regard to its user group. However, the target group of the explanation method are model developers, which are difficult to reach. Instead, a requirement of existing background knowledge in the field of machine learning is imposed on participants. Therefore, the target user group first has to be identified and briefed from the survey respondents before studying their comprehensibility. Consequently, the first part of the survey consists of a filtering question, where non-target group users (i.e., without any prior machine learning knowledge) are excluded from responding. The target users are briefed about the data and task used in this thesis in order to prevent confusion during the survey with regards to the data.

As detailed before, comprehensibility contains both the constructs of clarity and coherence. As these constructs are strongly overlapping, they cannot be viewed separately. Therefore, two types of questions are proposed to study both. In the first task, the users are shown three explanation examples. These are shown in random order to prevent any question ordering bias. The users are asked to explain their understanding of the model prediction as shown by the explanation in free text. These free-text answers are post-processed by three people with preknowledge in machine learning in an inductive clustering task of possible different interpretations. This is done independently from each other and without preknowledge of the targeted metric. The average number of clusters represents the clarity and coherence of the explanation. Explanations that are clear and unambiguous, as well as matching with the user's prior belief are hypothesised to leave little room for different user interpretations as they are understood in the same way. Thus, a small number of clusters, ideally one, is desirable as it indicates that the explanation is comprehensible. The instructions given to the annotators are provided in appendix E.

Secondly, a forward simulation task is selected [58]. Three new explanations are presented to the users. However, these miss the information about the model's prediction, as well as the confidence distribution. The users are asked to simulate the model behaviour given the explanation and predict the model classification. Additionally, they are asked to explain their simulation in free text. The idea behind this question is to study the comprehensibility of the explanation through an analysis of user understanding. This task also addresses both clarity and coherence, as explanations that are unambiguous and match the user's prior belief are hypothesised to be easily understandable. Thus, the model behaviour should become clear through the explanation, so that users can easily simulate the model. To measure this task, the classification accuracy of the users with regards to the true model predictions is computed. High accuracy is desirable. The provided reasoning for the user's predictions is considered in

addition to this metric, to give further insight into the user's understanding of the explanation. The full survey design is provided in the attachment.

6.2.2 Results and Discussion

The user study was conducted from 01/10/2021 to 11/10/2021 for a total of 11 days. Of the 36 respondents, three did not fulfil the selection criteria of prior background knowledge or experience in supervised machine learning. Thus, they were not presented with the survey questions and can be neglected in the analysis, leaving 33 responses for the qualitative analysis of human-comprehensibility of TSA.

Out of the 33 respondents, 26 (i.e., 79%) indicated that they understood the explanation components from the visualisation after the briefing. Two respondents also contacted the researcher to clarify the explanation components, but it is unclear at which point in the survey this occurred. Six respondents (i.e., 18%) said that they are not sure whether they fully grasp the explanation components. The main concern is related to the understanding of the feature attribution overlay to the data. Some participants were unsure about the different intensities of the feature attribution colour, while for another the connection of the colours to the labels was not instantly clear. Another participant indicated that their machine learning knowledge is limited and they were not familiar with the meaning of the bias anymore, while another was confused by the visualisation of the spiking data (i.e., the empty spaces in between the spikes). Furthermore, one respondent questioned the chosen dataset and task. They answered that they are not able to "understand what [the explanation for the prediction is]" because "there is nothing to explain [if a sensor seems to be directly connected to a certain class]". One respondent indicated that they did not understand the explanation components from the briefing, saying that they are unable to distinguish the colours used for the feature attribution. Moreover, participants were able to submit any comments they may have at the end of the survey. From these comments, the participant's insecurity and hesitance also becomes apparent as some participants mentioned that they potentially misunderstood the survey tasks and the topic. Some participants also mentioned directly to the researcher that they have never seen a local feature attribution-based explanation before, which caused uncertainty. As XAI is not necessarily a standard part of the machine learning curriculum, some confusion was expected. However, the briefing had the objective to clarify these confusions, where it was mostly, but not always sufficient.

During the analysis of the survey responses, these concerns have to be considered. The visualisation is not completely clear to some participants, which could potentially impact the way these respondents answered the rest of the survey, especially as human-comprehensibility is to be studied. In the following, the survey results regarding the user understanding questions as well as the simulation task are presented, analysed and discussed.

User Understanding

For the first part of the survey, three explanation examples were shown to the users in random order (namely timestamps 4918, 87083, 184970 of the test set) and users were asked to explain their understanding of why the SNN model made a certain prediction. Hence, qualitative data in form of free-text answers was collected in this part of the survey and clustered by three annotators into themes that shall represent different interpretations of participants. One response to a question was excluded from the analysis due to an apparent user error, in which the participant misread the data, leading to out of context interpretations. The original cluster themes identified per annotator can be found in appendix F and an overview of the number of clusters identified by annotator is presented in Table 6.4.

	Number of clusters		
Explanation	**Annotator A**	**Annotator B**	**Annotator C**
#1	5	6	6[13]
#2	5	10	9[13]
#3	3	8	7
Average	**4.33**	**8**	**7.33**

Table 6.4: Number of clusters identified by each annotator per explanation.

The number of clusters identified implies some ambiguity of the explanation as it exceeds a definite and clear interpretation. It shows that the explanation can be read differently by different users. The overall average number of clusters is at 6.55 which means there are on average more than 6 different interpretations of the model prediction given the visualised explanation, where one cluster likely corresponds to confusion by the explanation (i.e., the participants did not understand why the model made a certain prediction). However, many clusters are also close together (i.e., using the seat and the seat sensor being activated are considered separate clusters) and it can be argued that these slightly different perspectives refer to the same user understanding. Thus, the interpretations of these clusters are similar. Instead of different understandings, the clusters rather represent different parts of the explanation that the participants paid attention to.

As different labels between annotators are possible in an inductive approach, the granularity of themes differs per annotator, which can be seen by the number of themes and comments identified. While annotator A did not make a distinction between user action and sensor activation, annotators B and C did. In addition, annotator C also distinguished if a participant indicated uncertainty in their answer. In an effort to consolidate the annotations, we define common umbrella terms for the annotator themes. Namely, these are:

1. **Data**: This umbrella cluster combines the cluster themes that revolve around survey participants identifying the data itself as the reason for the SNN model prediction (i.e., *Only feature attribution* from annotator A, *sensor activation, use of x, colours, time* from annotator B and themes connected to sensor activation, user actions and colours from annotator C). This includes naming activated sensors that were displayed in the visualisation (e.g., “Because the cupboard sensor was activated”) as well as the user action inferred from the displayed data (e.g., “The cupboard was opened”). While the latter implies that the participant connected the data to user activities, both can be summarised in the common theme of **data**. From the survey responses, it is not clear whether participants always paid attention to the feature attribution map explanation, or whether they based their answer solely on the shown data and gave their reasoning from their domain knowledge. However, this cluster summarises the user responses that imply the reason for a model prediction to be found in the input data to the model. Most survey responses (61.6%) across all three explanations fall into this category.

2. **Classification confidence**: As a second, much smaller (8.8% umbrella cluster, responses to the question of why *TwoLayerSNN* made a certain decision indicate that survey participants related the model’s prediction to the shown classification confidence in the explanation visualisation (e.g. “Again, the confidence of the class Lunch outperforms the probability of other classes”). Participants mentioned the high classification confidence as the sole reason for the model prediction. However, participants did not indicate the

[13] Uncertainty themes if in combination with other identified themes were treated as comments, since the participant’s uncertainty still allowed them to give an answer to the question.

reason for the high classification confidence. Hence, it is unclear whether participants accepted the high confidence or perhaps thought about a deeper reason for the high confidence such as sensor stimulation or model bias, for example.

3. **Data and classification confidence**: In the third umbrella cluster, both preceding themes are combined. Survey respondents gave their understanding of the model behaviour by mentioning both the data as well as the model's classification confidence. This indicates that they viewed the explanation visualisation as a whole. This cluster is the second largest with 20.4% of the survey responses.

4. **Learned patterns**: This umbrella cluster consolidates survey responses that indicate the reason for the model prediction to be linked to the training process and learned patterns. Potential model bias is mentioned as well as possible dataset-specific information that may have been learned by the model, such as the time of day. It is very close to the **data** umbrella cluster, as learned patterns are recognised in the input data. Nevertheless, responses in this cluster incorporate the model training into their interpretation and understanding of model behaviour in addition. Thus, the responses in this cluster also show that the survey respondents looked at the explanation as well, and thought about the model behaviour. However, this was not often the case as 6.5% of the survey responses fall into this category.

5. **No certain answer**: The last umbrella cluster summarises the cases where the respondents were too confused by the visualised explanation to conclude reasoning for the model prediction. Even though this cluster is quite small, it is worthy to mention as it indicates that either the explanation or the task was not clear to the participant. This cluster is the smallest with 2.7% of the survey responses, showing that some participants were confused about some explanations but neither a certain explanation nor a certain participant can be identified as particularly confusing or confused.

To validate the above-explained umbrella clusters, the responses are coded to these themes using a mapping of the originally defined clusters by the annotators, and an assessment of the inter-rater reliability (IRR) is performed. As there are three raters, Fleiss' kappa [66] is taken as an IRR metric. As $\kappa = 0.592$, there is a moderate agreement between the raters. Thus, the mapping of the original annotations to the umbrella clusters makes sense and is in line with the original annotations. The mapping as well as the computation of κ can be found in the attachment.

These umbrella clusters represent the different themes that occurred in the survey responses. However, they do not necessarily match an answer to the question of *Why did the model make a certain prediction?*. Rather, these themes capture what respondents paid attention to while answering this part of the survey. While the **Learned patterns** cluster answers this question, it is quite small with 6.5% of the survey responses, thus corresponding to a small sample size.

An interesting observation is that some responses in the **data, confidence or both** clusters also refer to the attribution highlights (e.g. "high contribution of spiking seat sensor, in particular at later timesteps.", "feature attribution shows the sensors is of cupboard"). Additionally, the term *activation* could mean a spiking sensor, thus referring to the data itself, or alternatively the feature attribution highlights. Under the assumption that the mentioned sensor activation in the responses refers to attribution, most responses are based on the data as well as the attribution highlights. Thus, given this assumption, the explanations were mostly used in this survey part and led to the main interpretation of the shown explanations to be present in the input data.

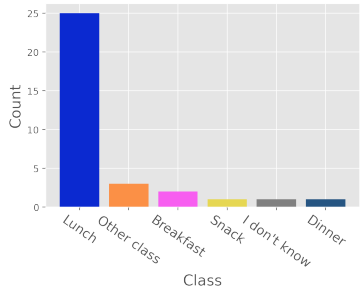
Furthermore, it is noteworthy that the survey responses indicate the occurrence of uncertainty within the participant's interpretations of model behaviour in the case of misclassification as all

answers in **No certain answer** correspond to the first explanation, which displayed a misclassification. This observation suggests that the participants do not intuitively understand why the model made a misclassification, and suggest that the coherence of the model predictions is relevant to the comprehensibility of the explanation. Especially the confidence graph seems to cause confusion and uncertainty in the human-comprehensibility of the explanation if it does not match the attribution shown in the explanation (e.g. “It is unclear why the confidence of Other is so high when the feature attribution graph does not clearly show this.”). This observation is also found in the responses to the second explanation, in which the model correctly classifies *Lunch*, but the feature attributions of the model are not strictly toward this class. As the attribution did not make sense and was incoherent with the respondents understanding of the data, the responses also showed hesitance (e.g., “[...]I am unsure (based on the feature attribution window), why the confidence distribution is spread out so broadly across other classes. [...]”). However, other responses suggested that the participants did have an understanding of non-zero confidence values of other classes (e.g. “[...]Small contribution of bias sensor to other labels still.”). In the last presented explanation, the model makes a correct classification with almost 100% confidence at the predicted class, which seemed to be very clear to the participants (e.g. “Major confidence, almost 100% certainty on the class Spare_Time/tv”). Therefore, the human-comprehensibility of the explanation is implied to be connected to the coherence of the model behaviour.

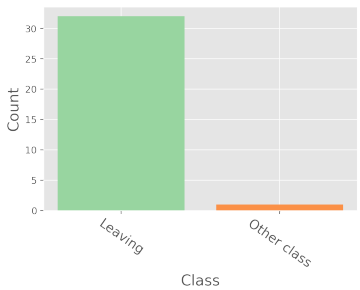
Simulation Task

Three examples were presented to the users for the forward simulation, namely of timestamps 81860, 5571 and 70169 of the test set. These examples showed the activities *Breakfast, Leaving, Sleeping* and were predicted by *TwoLayerSNN* as *Lunch, Leaving, Sleeping* respectively. Hence, two examples showed an explanation for a correct prediction of the model and one example showed an explanation for a misclassification.

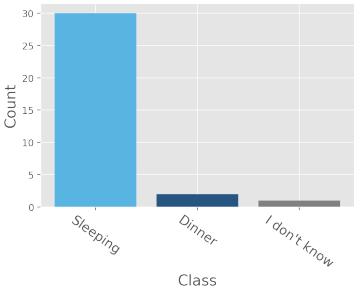
The sample of timestamp 70169 was correctly classified by *TwoLayerSNN* with strong confidence of 1. Also, the sample of timestamp 5571 was correctly classified with high confidence of about 0.85 for leaving. For the misclassified sample of timestamp 81860, the model predicted *Lunch* with confidence just below 0.5. The unmasked explanations can be found in appendix D. The predictions by the survey participants per question are displayed in Figure 6.9. Table 6.5 shows the resulting accuracy per question as well as the averaged accuracy over all simulation questions.



(a) User simulation predictions for question 1. The model prediction is *Lunch*, while the ground truth is *Breakfast*.



(b) User simulation predictions for question 2. The model prediction and ground truth is *Leaving*.



(c) User simulation predictions for question 3. The model prediction and ground truth is *Sleeping*.

Figure 6.9: Simulation task predictions per question.

In all cases, the majority of the participant’s prediction is the same as the model prediction,

Simulation Question	Accuracy
Question 1	0.758 ± 0.146
Question 2	0.970 ± 0.058
Question 3	0.909 ± 0.098
Averaged	$\mathbf{0.879 \pm 0.111}$

Table 6.5: Results of the user study simulation task (Accuracy of user predictions concerning model predictions reported at a 95% cI).

which is reflected in the mean user simulation accuracy of 0.879 ± 0.111. From all participants, 21 correctly predicted the model classification in all cases (i.e., 63.6% of participants). Across all examples, participants indicated that they chose their prediction mainly based on the feature attribution colour (e.g., “The only highlighted part in the image (besides the bias) is dark blue.”, “Only the green color is present [...].”, “Same color as sleeping. [...]”). Another often occurring reason given is the activation of a certain sensor/input dimension (e.g., “The cupboard is the only active sensor in the time before.”, “Maindoor is active.”, “Continuously sensor reading from the bed”‘). A few also mentioned the feature attributions in the bias to be relevant to their prediction (e.g., “The prediction is based on the attribution of bias sensor at second 81860.”), as well as the duration of the spike trains of the input (e.g., “The duration was too short for lunch / breakfast / dinner. Would guess snack because of that”). It is noteworthy that the lowest user simulation accuracy (0.758 ± 0.146) is achieved in the case of misclassification and lowest confidence of the samples by the model in the first question (see Table 6.5). This question also shows the largest variety of predicted classes, with five different predictions and one participant who was unable to make a prediction with the given information (see Figure 6.9a). For the second and third question, two classes were predicted each, where the third question also had one case where the participant was unable to decide on a class. However, the highest user simulation task accuracy does not correspond to the sample with the highest model confidence.

Even though the overall user simulation task accuracy is high (0.879 ± 0.111) and the feature attribution map overlaid to the data is stated by most participants as the main reason for their prediction, some respondents seemingly did not let the feature attribution influence their prediction. Therefore, a high level of clarity and coherence cannot be inferred by default. The results indicate that the explanation successfully gives insight into the model behaviour and can be understood by humans but at the same time is not straightforward. As some participants based their prediction on the data itself, rather than the data and feature attribution from the explanation, the high accuracy cannot be concluded to be completely due to the human-comprehensibility of the explanation. The nature of the dataset also contributes to the performance as the connection of certain sensors to certain activity classes are natural and make sense (e.g., *Bed* sensor activation is connected to *Sleeping*). However, this fact emphasises the coherence of the explanation since participants partly used the feature attribution and sensor activation as one unit (e.g. “Sleeping sensor is activated”). As the connection between the sensors and the model prediction is sound and coherent in the given example, referring to the attribution and sensor activation as one unit is possible. However, this is not true as a sensor does not necessarily have a one to one relationship with a certain class, which was interestingly suggested by one respondent (“I feel this was purely a one sensor to one outcome mapping. I am not able to understand how this is providing the explanation.”). In particular, the human-comprehensibility of the explanation suffers in case of misclassification. In the case of question one, the model predicted *Lunch* falsely, but the attribution toward this class was shown in the feature attribution. Since the participants knew the true label, the explanation was not coherent with their knowledge, which partly led to confusion. This confusion becomes ap-

parent in the responses, where other meal-related classes (i.e., *Breakfast, Snack*) were also predicted, next to *Lunch*. The responses to question two show that a feature attribution map that is evidently highlighting one particular class is clear and unambiguous as an explanation. Almost all participants correctly predicted the model behaviour. From the comments, it seems that the participant which did not predict *Leaving* was unable to see the feature attribution ("I am not really sure what happens when nothing was recorded but it would make sense to me that the prediction is other in that case."). Question three answers demonstrate that the participants paid attention to the feature attribution map in this task, as the two classes that were shown in the feature attribution map are both predicted. In summary, most participants were able to correctly predict the model behaviour, showing that the explanation is clear and quite unambiguous. Therefore, the results from the simulation task suggest that the explanation from TSA-S itself is human-comprehensible, with the clarity and coherence depending on the model predictive performance.

6.3 Implication and Outlook

In this chapter, an evaluation framework for testing the explainability of temporally coded SNN models with the help of TSA is deduced from related work in XAI in time series and applied. To conclude the findings of the evaluation, the main implications from the evaluation are shortly presented.

The tested qualities in the evaluation highlight different quality aspects. Therefore, dependencies of different metrics from the technical and user evaluation are not necessarily expected (i.e., a faithful explanation is not necessarily human-comprehensible). However, some connections can be observed between metrics and impacts connected to certain TSA properties can be observed recurrently in the analysis of the different metrics.

On the one hand, faithfulness and attribution sufficiency as metrics are connected in meaning. They both inspect how well TSA explanations reflect the model behaviour. Faithfulness and attribution sufficiency are prerequisites for a feature-based explanation to fulfil as they indicate whether model behaviour is reflected through the explanation. Nevertheless, they inspect different perspectives, so that it cannot be concluded that faithful explanations are also sufficient and vice versa. On the other hand, stability concerns the model behaviour to changed input specifically, and human-comprehensibility is a user-centred metric. Stability is desired because unstable explanations tend to be difficult to understand [63]. This motivation shows the connection between stability and human-comprehensibility. Certainty as a fulfilled property by the explanation is linked to the human because it provides additional information about the model prediction. Thus, stability, certainty and human-comprehensibility focus on qualities that specifically revolve around the user, where TSA-S outperforms TSA-NS. In this sense, the experiments with TSA show that the superior explanation method in faithfulness and attribution sufficiency is not necessarily a good explanation overall. TSA-NS is inferior in stability. A user study with TSA-NS was not conducted because the visualised explanation did not make sense. This highlights the importance of comprehensive evaluations in XAI as various quality aspects apply to a good explanation.

In the analysis of the evaluation results, particularly of the technical evaluation, certain properties of TSA are connected to its explanatory performance. First, the consideration of non-spikes as negative attribution (TSA-NS) or zero attribution (TSA-S) has a clear effect on all metrics. As described in the respective sections, the absence of spikes evidently carries relevant information for an SNN's prediction, which makes sense. Therefore, TSA-NS is superior in faithfulness and attribution sufficiency, which analyse the fundamental functionality of TSA as an explanation method. These metrics evaluate whether TSA successfully reflects the model behaviour.

Nevertheless, considering non-spikes as attributing as well also leads to larger explanations in size, which influences the stability of TSA. TSA-NS, which generates larger explanations in size, is less stable than TSA-S when it comes to natural perturbations because more features change in attribution values. Second, TSA is shown to explain shallow models better than deep models. Regardless of the TSA variant, the explanations extracted from *OneLayerSNN* generally achieve higher explanatory performance than the explanations extracted from deeper models. Hence, the layer propagation of weighted NCS' in the TSA algorithm requires improvements to enhance the applicability of TSA for deep SNNs. Lastly, the decay rate used in the computation of the NCS component of the TSA algorithm is implied to affect TSA's explanatory performance. While it makes sense that the decay rate used to compute spike contribution is the same as the SNN model's decay rate, the experimental results from the evaluation suggest that the decay rate impacted the explanatory performance of the TSA explanations. As this algorithmic design choice reflects the impact of a spike in an SNN model closely, a conclusion regarding the evaluation procedure can be formulated: The decay rate influences the size of the explanation (e.g., explanations from *TwoLayerSNN* with a larger decay rate are generally smaller than of *OneLayerSNN*), so that the evaluation of models with different decay rates may not be appropriate using the same maximum feature segment size. The assumption taken for the definition of the feature segment size is that the attribution values within a segment are close to each other. However, with a large decay rate, the attribution values change more rapidly, so that a smaller feature segment size is required to fulfil the assumption.

7 DISCUSSION

In this chapter, the research results which are presented in the previous chapters are reflected on the research questions that this thesis set out to answer. Additionally, limitations that must be considered in the frame of this thesis are highlighted.

7.1 Answer to Research Questions

This thesis set out to answer the research question of ***How can the predictions of temporally coded spiking neural networks be explained reliably?*** by studying the sub-questions *How can feature attribution be calculated for temporally coded spiking neural networks?* and *How can the quality of local feature attribution-based explanations extracted from SNNs be measured?* using a time series classification use case with a fully connected SNN with LIF neurons, temporal coding, as well as surrogate gradient learning.

A novel approach called *Temporal Spike Attribution* (TSA) is developed to answer the first sub-research question. TSA considers the spiking behaviour of the network S, learned weights W as well as the output layer membrane potential $U^{(L)}$. TSA is proposed as an algorithm to determine feature attribution in temporally coded SNNs. The algorithm builds on the use case problem as well as the use case SNN but applies to all temporally coded SNNs with static weights, as opposed to the model-specific explanations through feature strength functions presented by Jeyasothy et al. (2019) [37]. The components of the algorithm are defined as the NCS $\vec{N}_{t'}^{(l)}(t)$ which considers spike times t', normalised weight components $C_W(W^{(l)})$ and classification confidence $\vec{P}(t)$ based on $U^{(L)}$. Through addition and multiplication of the algorithm components, the model activation and forward propagation are approximated in two different variants: TSA-S only considers spikes of a neuron to affect its downstream neurons whereas TSA-NS also considers the absence of spikes. A particular input x's attributions to the different outputs $\hat{y}$ are computed through TSA. These represent and enable the inspection of the model behaviour, thus providing a local explanation in the form of feature attribution maps per class that addresses the outcome explanation and model inspection problem. Moreover, the model's certainty in its prediction is quantified as the softmax probabilities of the output layer. TSA applies the NCS presented by Kim and Panda (2021) [40] and extends them by incorporation of the learned weights of the model and the quantification of prediction certainty. Therefore, more information about the model behaviour is directly included in the explanation. In the TSA-NS variant, the temporal contribution definition is extended to non-spikes as well. Moreover, the exact spike times of the neurons in an SNN carries more meaning when using temporal coding as opposed to rate coding. Thus, this study showed the applicability of the NCS [40] in temporally coded SNNs on a time series classification task.

Furthermore, a thorough evaluation of TSA demonstrated on the time series classification use case is provided in the frame of this thesis to ensure the reliability of the explanation method and answer the second sub-research question. A technical and user evaluation covering the aspects of faithfulness, attribution sufficiency, sensitivity, certainty, and human-comprehensibility

is performed to measure the quality of TSA as well as to provide an evaluation framework for local feature attribution-based explanations on time series data. This evaluation is performed using three SNN models of different depths trained on a time series classification task, namely the activity of daily living using binary sensors recognition [50].

Faithfulness refers to the faithfulness of the explanation toward the model behaviour [42]. Hence, it is concerned with the question of whether the attributions of the explanation are the true attributions to the model prediction. As there is no explanation ground-truth to compare the attributions to, faithfulness is measured in explanation selectivity [59] in this thesis. By iterative inversion of the feature segment values with the highest attribution score and recording the model predictions, the effect of changing the value of the attributing feature segments is measured. The evaluation shows that TSA-NS in particular provides faithful explanations while TSA-S misses the attributions from non-spiking data and therefore does not outperform the baseline. Thus, this evaluation highlights the relevance of non-spikes to the model prediction which a faithful explanation method should consider. Additionally, TSA's improved faithfulness *OneLayerSNN* demonstrates the improved applicability to shallow SNN models. Hence, TSA may need to be adapted for deeper models to achieve better faithfulness.

Attribution sufficiency measures whether all truly relevant features are included in the explanation, thus whether the important features per explanation are sufficient for the model prediction [61]. The metric chosen in this thesis is the degree of sufficiency, which inspects the model performance when the background, i.e. non-important feature segments, are randomly shuffled within their input dimension. Different thresholds for considering attributions as important are tested, where a high degree of sufficiency is only reached with a threshold of 0. Therefore, TSA does not distinguish between important and non-important attributions with regard to sufficiency. TSA-NS showed particularly high attribution sufficiency with this threshold, supporting the observation from the faithfulness experiments that the non-spiking parts of the input attribute actively to the prediction and should not be neglected in a feature attribution-based explanation.

Hence, the experiments testing faithfulness and attribution sufficiency suggest that TSA successfully reflects the model behaviour of the SNN models to be explained, where TSA generates better explanations for shallow models as well as under consideration of non-spiking input. These results indicate that TSA as an explanation method is effective in capturing SNN model behaviour, which is essential to an explanation method.

Sensitivity describes an explanation method's desired ability to generate similar explanations for similar input. Stability (i.e., sensitivity to naturally occurring data perturbations) and robustness (i.e., adversarial perturbations) are distinguished [48], where the latter is not evaluated in this thesis. Sensitivity is measured in max-sensitivity [63], and realised as the Frobenius norm of the difference in explanations with clean and perturbed input. Both implementations of TSA achieve a high level of stability across the different tested SNN models, where TSA-S outperforms TSA-NS. The decay rate of the membrane potential likely influences the stability. As TSA is a deterministic algorithm, it also fulfils the property of consistency, meaning the generation of the same explanations for the same input.

Moreover, the certainty of the model prediction is provided in the explanation as well as considered in the TSA computation. Information about certainty in the explanation can put the explanation attributions into context [48].

Lastly, human-comprehensibility concerns itself with the target group of the explanation and focuses on how understandable an explanation is to its target users. To evaluate this, studies with humans are required [6]. In the frame of this thesis, a user study in the form of a survey was conducted to qualitatively evaluate human-comprehensibility through the constructs of clarity and coherence for the TSA-S explanations as the TSA-NS explanations are deemed incoherent beforehand. Two tasks were presented to the users. The first task evaluated the

user interpretations of explanations and data, and the answers were clustered into themes of different interpretations by three coders. In the second task, users were asked to simulate the model prediction given the explanation [58]. The accuracy of the simulation is then the metric to measure the comprehensibility by. The results of the user study suggest that the explanations generated by TSA-S are clear in showing what features attribute to the model prediction, but it is not always coherent. This is especially the case of misclassification, or if the data itself is incoherent with the user's beliefs. Furthermore, the interpretation task showed that different participants focus on different parts of the explanation during interpretation. Therefore, TSA-S can be human-comprehensible but this quality is also tightly linked to the predictive performance of the SNN model to be explained.

Thus, the evaluation framework covers different aspects of a good feature attribution explanation on time series data and can be transferred to other local feature-based explanations on time series tasks. Concrete metrics are provided so that such explanations can be benchmarked in possible future work. While the framework is quite comprehensive, the evaluation shows that it is not complete. For example, the number of non-zero attributing features is implied to contribute to explanatory performance, especially concerning human-comprehensibility. However, this was not evaluated. Similarly, robustness was not evaluated because it requires other questions to be solved first. Therefore, this framework represents a first comprehensive evaluation methodology for TSA explanations that addresses the second sub-research question of this thesis and has the potential for extension and improvement. This topic is reflected on in dedication in the upcoming section 7.2.

In summary, this thesis defines a way to explain the predictions of temporally coded spiking neural networks through the definition and evaluation of *Temporal Spike Attribution* quite reliably where TSA-S and TSA-NS both have their strengths and weaknesses. Overall, observations from this study suggest that TSA is better at explaining shallow models rather than deeper models which highlights the need to adapt the computation of attributions in deeper models. In addition, insight into the attribution of spikes and non-spikes of the data is gained where the non-spiking input should not be neglected. As both these topics are interesting to inspect, they are reflected separately in the following sections 7.3 and 7.4.

7.2 Reflection on Evaluation Framework

The target of the evaluation is to provide a thorough evaluation of local feature-based explanations on time series data such as TSA. The technical evaluation spans the explanation qualities of faithfulness measured in explanation selectivity [59], attribution sufficiency measured in the attribution sufficiency score [61], and stability measured in max-sensitivity [63]. No new metrics are introduced in this work, rather existing metrics that match the data and explanation format are applied. An evaluation of robustness is not performed in the frame of this thesis because the definition of adversarial robustness for explanations must be clarified first. Therefore robustness is still missing for a complete evaluation of TSA's sensitivity. Additionally, the size of the explanation is not evaluated directly. Still, it indirectly impacted the other explanation qualities (e.g. TSA-NS explanations are larger because they consider non-spiking attribution, which impacted the max-sensitivity scores). A dedicated metric could be introduced to measure the explanation size because it is connected to human understanding [58]. However, the optimal size is not straightforward: a smaller size does not necessarily lead to a better explanation because it may be missing information (e.g., TSA-S explanations are smaller but also perform worse in faithfulness and sufficiency than TSA-NS explanations). Instead, a balance of size and informativeness in the explanation has to be reached for which a metric has to be proposed.

In the user evaluation, only human-comprehensibility in terms of clarity and coherence is tested.

This is a quality of the explanation that requires a human-grounded evaluation. A user study design with free-text user understanding questions and a forward simulation task [58] is performed to assess the comprehensibility of TSA-S explanations. The free-text answers are clustered inductively by three people which are not part of the research team without awareness of the target metric, i.e., the number of clusters as an indicator for clarity. While this approach prevents a researcher bias toward good results, the clusters rather capture which part of the explanation users paid attention to instead of the understanding. The instructions and definition of the clustering task, which took place remotely to limit the researcher influence, might be unclear. Therefore, future studies using this methodology shall take care to improve on this aspect.

7.3 Reflection on Explaining Deep Models

The performance of TSA as a local explanation method is tested on different SNN models in this thesis. These models have the same neuron model and neural coding and were trained using the same learning method. However, the number of layers is different. In the experiments to evaluate TSA as an explanation method, a trend showed toward better performance for *OneLayerSNN*, which is the shallow model without any hidden layers. TSA performed best in all quantitative metrics across the variants, which suggests that it is better at generating explanations for shallow models than for deep models.

In the TSA algorithm, the difference in depth means that multiple weighted NCS are multiplied and aggregated in the input space for deep models, while shallow models are explained using the weighted NCS score of the input spikes. Therefore, the multiplication of multiple weighted NCS scores may not fully capture the complexity of a hidden layer in the SNN model to be explained. The computations do not seem to approximate the model behaviour as well as in the case of shallow models. The reason for this is unclear, however, the size of the hidden layers, as well as the number of hidden layers, likely contribute to this. The size of the hidden layer is connected to the sizes of the matrices of the TSA computation. In the aggregation of values in the input space to arrive at attribution values, the hidden layer sizes influence the computation in the sense that larger hidden layers generally result in larger attribution values due to the additive operation of matrix multiplication. At the same time, the number of hidden layers also influences the attribution values. As the possible values for each component of the computation (i.e., NCS of S, weight contribution of W, outcome probability computed from $U^{(L)}$) are realised in the interval $[0, 1]$, TSA could be at risk of vanishing attribution values for deep models due to repeated multiplication of values smaller than 1. Consequently, the information extracted for the explanation could be blurred by the number of hidden layers and distorted by the hidden layer sizes.

To solve this issue, further research into the TSA algorithm is required. Some form of normalisation could counteract the effect of the hidden layer sizes, while measures to prevent vanishing values, such as operations in log-space or intermediate normalisation of weighted NCS for example, could be implemented to prevent this issue. Nevertheless, these suggestions need to be studied and drawbacks considered (e.g. negative values are undefined in log-space).

7.4 Reflection on Non-Spiking Attribution

In the frame of this research, TSA explanations were tested in two variants which are distinguishable by the way they consider non-spikes in their computation. This is realised in the spike time component of the TSA algorithm, namely the neural contribution score (NCS) (see

section 5.2.1). On the one hand, TSA-S defines the contribution of a neuron i to its downstream neurons at time t' as zero in case of no spike because there is no effect of $x_{i,t'}$ on the downstream neurons. Instead, a downstream neuron j's membrane potential u_j decays according to the LIF dynamics. In TSA-NS, on the other hand, the absence of a spike at t is assumed to affect the downstream neurons in a sense that u_j is neither decreased nor increased by a postsynaptic potential. Therefore, this effect is captured by the NCS as the negative of a spike's attribution.

Hence, the explanations that are extracted with TSA-S and TSA-NS are different, as the experimental setup and results show. Not only does their size differ, as TSA-NS considers a larger number of features, but their explanatory performance is also different. While both methods exceed the baseline in all tested properties, there are noticeable improvements of TSA-NS over TSA-S in terms of faithfulness to the explained model and sufficiency of the attributions to the model prediction. Both of these properties describe that TSA-NS is better at capturing model behaviour in its explanations. This suggests that the models use not only the spiking part of the input but also the absence of spikes as information for the prediction. Therefore, the interpretation of spike absence as zero is wrong as it does not capture the full model behaviour in the explanation. Nevertheless, TSA-S achieves better stability and offers human-comprehensible explanations, which TSA-NS does not. The TSA-NS explanations, however, are incoherent with the data, ground truth and model prediction when observing class activation in the explanation. In the case of the ADL classification task, the sensor activation related to an activity attributes to what seems to be random classes but the predicted class. However, information is carried in spikes in temporal coding [17], TSA-NS must be wrong regarding the attribution strength it assigns to features. Since the only difference between both TSA variants is the definition of the NCS in case of spike absence, an incorrect definition of the NCS for absent spikes is implied.

Therefore, a need to refine the definition of the NCS concerning non-spikes is highlighted. In the experiments of this thesis, the NCS of non-spikes in TSA-NS is defined as $-\exp(\gamma|t-t'|)$, where γ is the membrane potential decay rate dictated by LIF neuron dynamics, t is the time of prediction and t' is the time of the non-spike. It is the negative version of the NCS for spikes, in other words, it is the NCS with a weight of -1. Thus, the weight definition likely requires improvement for more coherent explanations. A future study could explore different approaches with smaller weights depending on the data. An option could be to compute the NCS relative to the amount of input dimensions, i.e. $N_{i,t'}(t) = -\frac{1}{D}\exp(\gamma|t-t'|)$ if $x_{i,t'} = 0$. This approach could capture the relative meaning of a single non-spike in a single dimension in the context of the input data. Another option could be to examine the input at t' and define the attribution relative to the spiking input at t'. I.e., $N_{i,t'}(t) = -\frac{1}{M}\exp(\gamma|t-t'|)$ if $x_{i,t'} = 0$ where M is the number of dimensions which do not spike at t'. Such an approach would emphasise overall the importance of absent spikes in certain dimensions, while other input dimensions are spiking. Moreover, the attribution of a non-spike could be defined by how much the membrane potentials of the downstream neurons decay in the time step. This definition would capture the effect of a non-spike from a different perspective: Instead of considering that the downstream neuron's membrane potential was not changed by the postsynaptic potential, it looks at the true evolution of the downstream neuron's membrane potentials. However, these suggestions are only ideas so far, which have to be properly researched in future work to validate them.

7.5 Limitations

While this research can provide answers to the posed research questions, there are several limitations to consider when interpreting the results.

First of all, the use case models and data represent a specific use case on which the research

is based. Even though TSA is designed to be model-agnostic for temporally coded SNNs and is not specific to the use case, it has only been demonstrated on the models and data in this thesis. SNNs with other neural models like the Izhekevich neuron [20] for example, were not tested. Further experiments with other SNN architectures are therefore required to validate the findings of this thesis for a larger class of SNN models. From the data perspective, a similar limitation applies. As the experiments solely looked at the binary ADL dataset, TSA is strictly speaking tested with binary sensor data alone, which was directly translated into spikes. Hence, the dataset requires little effort in neural coding. Another type of time series data which is not limited to binary values requires neural coding to spike trains before explanations could be extracted using TSA. Also, reverse coding would have to be studied to relate the spike code as well as the feature attribution extracted based on the spike code to the input space. The choice of the binary ADL dataset is therefore convenient for the scope of the research. However, to generalise the findings of this thesis to applicability on other datasets and tasks, studies with more complex time series data and other data types would be required that include neural coding and decoding.

Additionally, the implemented models in this work are not optimally performing in terms of predictive accuracy, runtime as well as memory usage. The models do not reach comparable predictive performance on the dataset as recent work [54], thus the explanation may not make sense at times. This especially could pose a limitation for the evaluation of human-comprehensibility as incoherent model behaviour becomes likely. Also, the models require quite a lot of memory as they record the history of their neuron's membrane potentials and spike trains from $t = 0$, and the sequential processing leads to long runtimes. Hence, assumptions that enable a certain degree of parallelisation were made during model training and explanation extraction. Therefore, the inherent temporal processing of SNNs is limited during model training. However, as the models were able to learn, this is a minor point. For the explanation, however, potentially some information could be lost by only considering the last hour before the time point that is explained. Moreover, the test set for quantitative evaluation is rather small due to reasons of computational efficiency as well, so that the generalisation of the results is limited.

There are also some limitations in the TSA computation. TSA explanations provide an attribution map per class so that the input's attribution to a certain output can be inspected individually. At the same time, this limits the explanation as the information becomes complex with a growing number of classes, and cross-class effects in the attributions are not specifically identified. These could be interesting in an explanation however as they could contribute to understanding the logic behind a model's prediction. Moreover, the computation is based on matrix multiplications, which are essentially multiplication and addition operations of the matrix elements. As mentioned in section 7.3, this leads to limitations regarding deep models connected to the number of hidden layers and their size. A considerable limitation to the computation of TSA-S is the NCS, which is essentially only computed for the spiking part of the input as mentioned in the previous section. Consequently, the non-spiking parts of the input do not exhibit attribution values, even though these carry relevant information to the model prediction as shown by the evaluation of faithfulness and attribution sufficiency. However, TSA-NS does not seem to generate coherent explanations as it often weighs spiking input less than non-spiking input, even though they may seem faithful and sufficient to the model. Weighing the non-spikes with the same absolute value as spikes may not be the correct implementation.

Furthermore, there are considerations to take into account in the evaluation of the explanation as well. As mentioned during the discussion of the quantitative results, the random baseline used in this research may be a particularly bad performing baseline for the experiments. However, as the evaluation of XAI methods is not straightforward in terms of metrics and methods [6], a random baseline is an appropriate start to compare an explanation method's explanatory performance to. In addition, only the feature attribution map of the predicted class is evaluated

in the quantitative evaluation. As this represents only part of the explanation, the evaluation is only valid for the feature attribution map of the predicted class and it would be interesting to consider the whole explanation in further studies.

The small number of explanations shown in the qualitative evaluation (i.e., three per task) also limited the explanations that could be shown to participants, even though a balance of overwhelming a survey respondent and the survey content has to be ensured. However, it could also have been interesting to compare the explanations of the different SNN models of this study for example. In this research, the results of the qualitative study are based solely on the explanations extracted from *TwoLayerSNN* with TSA-S. Besides, a study regarding the optimal visualisation of the explanation to the user may have been needed before an evaluation of human-comprehensibility. Such a study would ensure the separation of the evaluation of explanation content, which was the objective in this thesis, and the evaluation of the presentation itself. Instead, a few users were confused by the selected colours or visualisation of the input data and spikes, which probably influenced their judgement. This confusion is apparent in the answered comments, as well, and suggests some underlying misunderstandings in the user study responses. Moreover, it becomes apparent in the survey responses that users thought about the explanation in different depths, leading to different abstraction levels in the answers. In retrospect, an in-place experimental setup such as a focus group or individual interviews for example might be a better choice as they allow the user to pose questions if there is confusion and clarify any prior questions directly with the researcher before the evaluation, and it allows the researcher to follow up on unclear answers.

8 CONCLUSION AND FUTURE WORK

This thesis presents *Temporal Spike Attribution* (TSA) as a novel local, post-hoc explanation method for temporally coded spiking neural networks that provides explainability through feature attributions in the input space. It is based on the model internals of SNNs, namely the spike trains of the neurons, the model weights, and the classification confidence. These components are generally found in SNNs so that TSA is an SNN model-agnostic explanation method. Two TSA variants, namely TSA-S which only considers spiking input as attributing, and TSA-NS which also considers non-spikes to carry information, are explored in this research. TSA is demonstrated on the activities of daily living with binary sensors dataset [50] using temporally coded SNNs of varying depths that employ LIF neurons, temporal coding and were trained using surrogate gradient learning. Using this use case, both TSA variants are thoroughly evaluated in faithfulness, attribution sufficiency, stability, certainty, and human-comprehensibility. The results validate TSA explanations as faithful, sufficient, and stable. While TSA-S explanations are more stable, TSA-NS explanations are superior in faithfulness and sufficiency, which suggests relevant information for the model prediction to be in the absence of spikes as well. Certainty is provided in both variants, and the TSA-S explanations are largely human-comprehensible where the clarity of the explanation is linked to the coherence of the model prediction. TSA-NS, however, seems to assign too much attribution to non-spiking input, leading to incoherent explanations and highlighting the need to research the relation between spike attribution and non-spike attribution. Furthermore, the applicability of TSA is best on shallow models without hidden layers which emphasises that further research is required to ensure the same level of explanatory performance of TSA on SNN models with hidden layers as well.

Even though TSA provides a method to explain the predictions of an SNN, there is a need for future research to address the limitations of this thesis as well as explore related fields. First of all, the attribution of non-spikes could be improved by studying the contributions of absent spikes to model predictions. Furthermore, TSA could be studied with other SNN models that are implemented in an SNN simulator, or even in neuromorphic hardware. Research in this direction is required to validate the model-agnostic property of the method. Research to improve the TSA algorithm could look into computation in logarithmic space to prevent vanishing attribution values for deep models. However, the algorithm requires some changes, as negative values, which occur due to negative weight contributions, are undefined in log-space.

Moreover, the evaluation framework presented in this thesis can be extended in future work. For example, explanation sensitivity requires further research and a more detailed definition. While stability shall measure the explanation's sensitivity to naturally occurring perturbations, the tested perturbations are still generated artificially, thus imitating natural perturbations. Future research could employ generative adversarial networks to create similar data within the distribution for a better approximation of naturally occurring perturbations. TSA is not evaluated for the robustness of the explanations with adversarial examples as well. This is an interesting field for future research, which is not limited to explanations of SNNs but all opaque models: the definition of adversarial robustness, as well as an evaluation framework for robustness, could be defined.

Future research in the field of optimal feature attribution visualisation is also required to improve the human-comprehensibility of such explanations. Especially in cases where the data is not inherently readable like in image data, feature attribution explanations may require additional information to enhance comprehensibility. Such studies could consider extending and enhancing the current visualisation by making it interactive, for example by enabling the user to investigate different time steps in order to understand the evolution of feature attribution over time better.

Furthermore, TSA offers a foundation for further forms of explanation methods. The attribution scores defined through TSA could be used in a temporal saliency rescaling method which finds saliency scores based on the occlusion of input and temporal dimensions [41]. Using such a method, the attribution of non-spiking input could be determined in a better way than with TSA-NS, too. Another worthwhile research direction could be utilizing TSA to generate counterfactual explanations, which are generally easy for humans to comprehend [36].

REFERENCES

[1] Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4), December 2016.

[2] Xolani Dastile, Turgay Celik, and Moshe Potsane. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, 91:106263, 2020.

[3] Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, 2001.

[4] Geoffrey Currie and K Elizabeth Hawk. Ethical and legal challenges of artificial intelligence in nuclear medicine. *Seminars in Nuclear Medicine*, 51(2):120–125, 2021. Artificial Intelligence in Nuclear Medicine.

[5] Jianxing He, Sally L. Baxter, Jie Xu, Jiming Xu, Xingtao Zhou, and Kang Zhang. The practical implementation of artificial intelligence technologies in medicine. *Nature medicine*, 25(1):30–36, 2019.

[6] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning – a brief history, state-of-the-art and challenges. In *ECML PKDD 2020 Workshops*, pages 417–431, Cham, 2020. Springer International Publishing.

[7] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[8] High-Level Expert Group on AI. Ethics guidelines for trustworthy ai. Report, European Commission, Brussels, April 2019.

[9] European Union. General data protection regulation, 2018. `https://gdpr.eu/tag/gdpr/`.

[10] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):1–42, 2019.

[11] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[12] Alan F. Murray. *Pulse-Based Computation in VLSI Neural Networks*, pages 87—109. MIT Press, Cambridge, MA, USA, 1999.

[13] H. Hagras, A. Pounds-Cornish, M. Colley, V. Callaghan, and G. Clarke. Evolving spiking neural network controllers for autonomous robots. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 5, pages 4620–4626, 2004.

[14] Tiffany Hwu, Jacob Isbell, Nicolas Oros, and Jeffrey Krichmar. A self-driving robot using deep convolutional neural networks on neuromorphic hardware. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 635–641, 2017.

[15] Mostafa Rahimi Azghadi, Corey Lammie, Jason K. Eshraghian, Melika Payvand, Elisa Donati, Bernabé Linares-Barranco, and Giacomo Indiveri. Hardware implementation of deep network accelerators towards healthcare and biomedical applications. *IEEE Transactions on Biomedical Circuits and Systems*, 14(6):1138–1159, 2020.

[16] Saima Sharmin, Priyadarshini Panda, Syed Shakib Sarwar, Chankyu Lee, Wachirawit Ponghiran, and Kaushik Roy. A comprehensive analysis on adversarial robustness of spiking neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[17] Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition / Wulfram Gerstner, Werner M. Kistler, Richard Naud, Liam Paninski*. Cambridge University Press, Cambridge, 2014.

[18] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

[19] Xiangwen Wang, Xianghong Lin, and Xiaochao Dang. Supervised learning in spiking neural networks: A review of algorithms and evaluations. *Neural networks : the official journal of the International Neural Network Society*, 125:258–280, 2020.

[20] E. M. Izhikevich. Simple model of spiking neurons. *IEEE transactions on neural networks*, 14(6):1569–1572, 2003.

[21] Wulfram Gerstner and Werner M. Kistler. *Spiking neuron models: Single neurons, populations, plasticity / Wulfram Gerstner, Werner M. Kistler*. Cambridge University Press, Cambridge, 2002.

[22] Wulfram Gerstner. Spike-response model. *Scholarpedia*, 3(12):1343, 2008.

[23] Zihan Pan, Jibin Wu, Malu Zhang, Haizhou Li, and Yansong Chua. Neural population coding for effective temporal classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[24] Sander M. Bohte, Joost N. Kok, and Han La Poutré. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48(1-4):17–37, 2002.

[25] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006.

[26] Peter U. Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9:99, 2015.

[27] Bodo Rueckauer and Shih-Chii Liu. Conversion of analog to spiking neural networks using sparse temporal coding. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018.

[28] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

[29] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[30] Friedemann Zenke and Tim P. Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural computation*, pages 1–27, 2021.

[31] Dan Goodman and Romain Brette. Brian: a simulator for spiking neural networks in python. *Frontiers in neuroinformatics*, 2:5, 2008.

[32] Hananel Hazan, Daniel J. Saunders, Hassaan Khan, Devdhar Patel, Darpan T. Sanghavi, Hava T. Siegelmann, and Robert Kozma. Bindsnet: A machine learning-oriented spiking neural networks library in python. *Frontiers in neuroinformatics*, 12:89, 2018.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[34] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[35] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[36] Christoph Molnar. *Interpretable Machine Learning*. 2019. `https://christophm.github.io/interpretable-ml-book/`.

[37] Abeegithan Jeyasothy, Suresh Sundaram, Savitha Ramasamy, and Narasimhan Sundararajan. A novel method for extracting interpretable knowledge from a spiking neural classifier with time-varying synaptic weights. *CoRR*, abs/1904.11367, 2019.

[38] Abeegithan Jeyasothy, Suresh Sundaram, and Narasimhan Sundararajan. Sefron: A new spiking neuron model with time-varying synaptic efficacy function for pattern classification. *IEEE transactions on neural networks and learning systems*, 30(4):1231–1240, 2019.

[39] S. Song, K. D. Miller, and L. F. Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3(9):919–926, 2000.

[40] Youngeun Kim and Priyadarshini Panda. Visual explanations from spiking neural networks using inter-spike intervals. *Scientific Reports*, 11(1), 2021.

[41] Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6441–6452. Curran Associates, Inc., 2020.

[42] Kevin Fauvel, Véronique Masson, and Élisa Fromont. A performance-explainability framework to benchmark machine learning methods: Application to multivariate time series classifiers. In *Proceedings of the IJCAI-PRICAI 2020 Workshop on Explainable AI (XAI)*, 2020.

[43] Emre Ates, Burak Aksar, Vitus J. Leung, and Ayse K. Coskun. Counterfactual explanations for multivariate time series. In *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*, pages 1–8, 2021.

[44] Ferdinand Küsters, Peter Schichtel, Sheraz Ahmed, and Andreas Dengel. Conceptual explanations of neural network prediction for time series. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2020.

[45] Taro Kono, Satoshi Yamaguchi, and Tomoharu Nagao. Time series prediction with dual reliability: Uncertainty and explainability. In Mohamed Elgendi and Hamed Shah-Mansouri, editors, *2020 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4095–4102. IEEE, [Piscataway, NJ], 2020.

[46] Roy Assaf, Ioana Giurgiu, Frank Bagehorn, and Anika Schumann. Mtex-cnn: Multivariate time series explanations for predictions with convolutional neural networks. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 952–957. IEEE, 2019.

[47] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[48] Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz Rodríguez. Explainable artificial intelligence (XAI) on timeseries data: A survey. *CoRR*, abs/2104.00950, 2021.

[49] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. `http://archive.ics.uci.edu/ml`.

[50] Fco Javier Ordóñez, Paula de Toledo, and Araceli Sanchis. Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors (Basel, Switzerland)*, 13(5):5460–5477, 2013.

[51] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124, 2010.

[52] Lawrence Mosley. *A balanced approach to the multi-class imbalance problem*. 2013.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[54] Rebeen Ali Hamad, Masashi Kimura, Longzhi Yang, Wai Lok Woo, and Bo Wei. Dilated causal convolution with multi-head self attention for sensor human activity recognition. *Neural Computing and Applications*, 2021.

[55] Fred Matthew Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE transactions on visualization and computer graphics*, 2018.

[56] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[57] Connor C. Gramazio, David H. Laidlaw, and Karen B. Schloss. Colorgorical: creating discriminable and preferable color palettes for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[58] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*, 2017.

[59] Grégoire Montavon, W. Samek, and K. Müller. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.*, 73:1–15, 2018.

[60] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A. Keim. Towards a rigorous evaluation of xai methods on time series. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4197–4201, 2019.

[61] Gabriëlle Ras, Marcel van Gerven, and Pim Haselager. *Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges*, pages 19–36. Springer International Publishing, Cham, 2018.

[62] Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, 2019.

[63] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in)fidelity and sensitivity of explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[64] Raymond S. Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of General Psychology*, 2(2):175–220, 1998.

[65] Hendrik Müller, Aaron Sedley, and Elizabeth Ferrall-Nunge. Survey research in hci. In Judith S. Olson and Wendy A. Kellogg, editors, *Ways of Knowing in HCI*, pages 229–266. Springer New York, New York, NY, 2014.

[66] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[67] Julian Büchel, Dmitrii Zendrikov, Sergio Solinas, Giacomo Indiveri, and Dylan R. Muir. Supervised training of spiking neural networks for robust deployment on mixed-signal neuromorphic processors. *CoRR*, abs/2102.06408, 2021.

[68] John J. M. Reynolds, James S. Plank, Catherine D. Schuman, Grant R. Bruer, Adam W. Disney, Mark E. Dean, and Garrett S. Rose. A comparison of neuromorphic classification tasks. In Thomas E. Potok and Catherine Schuman, editors, *Proceedings of the International Conference on Neuromorphic Systems*, pages 1–8, New York, NY, USA, 2018. ACM.

[69] S. Dora, K. Subramanian, S. Suresh, and N. Sundararajan. Development of a self-regulating evolving spiking neural network for classification problem. *Neurocomputing*, 171:1216–1229, 2016.

[70] Abdulrazak Yahya Saleh, Siti Mariyam Shamsuddin, and Haza Nuzly Abdull Hamed. Multi-objective differential evolution of evolving spiking neural networks for classification problems. In Richard Chbeir, Yannis Manolopoulos, Ilias Maglogiannis, and Reda Alhajj, editors, *Artificial Intelligence Applications and Innovations*, volume 458 of *IFIP Advances in Information and Communication Technology*, pages 351–368. Springer International Publishing, Cham, 2015.

[71] B. Chandra and K. V. Naresh Babu. Classification of gene expression data using spiking wavelet radial basis neural network. *Expert Systems with Applications*, 41(4):1326–1330, 2014.

[72] John J. Wade, Liam J. McDaid, Jose A. Santos, and Heather M. Sayers. Swat: a spiking neural network training algorithm for classification problems. *IEEE transactions on neural networks*, 21(11):1817–1830, 2010.

[73] Jianguo Xin and M. J. Embrechts. Supervised learning with spiking neural networks. In *IJCNN'01*, Piscataway, N.J., 2001. IEEE.

[74] J. E. Smith. A temporal neural network architecture for online learning. *ArXiv*, abs/2011.13844, 2020.

[75] Xingyu Yang, Mingyuan Meng, Shanlin Xiao, and Zhiyi Yu. Spa: Stochastic probability adjustment for system balance of unsupervised snns. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6417–6424, 2021.

[76] Alberto Patiño-Saucedo, Horacio Rostro-Gonzalez, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. Event-driven implementation of deep spiking convolutional neural networks for supervised classification using the spinnaker neuromorphic platform. *Neural networks : the official journal of the International Neural Network Society*, 121:319–328, 2020.

[77] Ting-Ying Zheng, F. A.N. Li, Xue-Mei Du, Yang Zhou, N. A. Li, and Xiao-Feng Gu. Unsupervised image classification with adversarial synapse spiking neural networks. In *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, pages 162–165. IEEE, 2019.

[78] Hui Liang, Jianxing Wu, Ran Wang, Feng Liang, Li Sun, and Guohe Zhang. A spiking neural network for visual color feature classification for pictures with rgb-hsv model. In *2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE)*, pages 36–39. IEEE, 2019.

[79] Jiaxing Liu, Hong Huo, Weitai Hu, and Tao Fang. Brain-inspired hierarchical spiking neural network using unsupervised stdp rule for image classification. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pages 230–235, New York, NY, USA, 2018. ACM.

[80] Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Stdp-based unsupervised feature learning using convolution-over-time in spiking neural networks for energy-efficient neuromorphic computing. *ACM Journal on Emerging Technologies in Computing Systems*, 14(4):1–12, 2018.

[81] Kourosh Kiani and Elmira Mohsenzadeh Korayem. Classification of persian handwritten digits using spiking neural networks. In *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pages 1113–1116. IEEE, 2015.

[82] Yu Wang, Tianqi Tang, Lixue Xia, Boxun Li, Peng Gu, Huazhong Yang, Hai Li, and Yuan Xie. Energy efficient rram spiking neural network for real time classification. In Alex K. Jones, Hai Li, Ayse K. Coskun, and Martin Margala, editors, *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, pages 189–194, New York, NY, USA, 2015. ACM.

[83] Hyeryung Jang and Osvaldo Simeone. Multi-sample online learning for spiking neural networks based on generalized expectation maximization. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4080–4084, 2021.

[84] Ali Rasteh, Floriane Delpech, Carlos Aguilar Melchor, Romain Zimmer, Saeed Bagheri Shouraki, and Timothée Masquelier. Encrypted internet traffic classification using a supervised spiking neural network. *ArXiv*, abs/2101.09818, 2021.

[85] Anand Kumar Mukhopadhyay, Atul Sharma, Indrajit Chakrabarti, Arindam Basu, and Mrigank Sharad. Power-efficient spike sorting scheme using analog spiking neural network

classifier. *ACM Journal on Emerging Technologies in Computing Systems*, 17(2):1–29, 2021.

[86] Clarence Tan, Marko Šarlija, and Nikola Kasabov. Neurosense: Short-term emotion recognition and understanding based on spiking neural network modelling of spatio-temporal eeg patterns. *Neurocomputing*, 434:137–148, 2021.

[87] Zhanglu Yan, Jun Zhou, and Weng-Fai Wong. Energy efficient ecg classification with spiking neural network. *Biomedical Signal Processing and Control*, 63:102170, 2021.

[88] Andrey V. Andreev, Mikhail V. Ivanchenko, Alexander N. Pisarchik, and Alexander E. Hramov. Stimulus classification using chimera-like states in a spiking neural network. *Chaos, Solitons & Fractals*, 139:110061, 2020.

[89] Moses Apambila Agebure, Elkanah Olaosebikan Oyetunji, and Edward Yellakuor Baagyere. A three-tier road condition classification system using a spiking neural network model. *Journal of King Saud University - Computer and Information Sciences*, 2020.

[90] Carlos D. Virgilio G, Juan H. Sossa A, Javier M. Antelis, and Luis E. Falcón. Spiking neural networks applied to the classification of motor tasks in eeg signals. *Neural networks : the official journal of the International Neural Network Society*, 122:130–143, 2020.

[91] Andrey Andreev and Alexander Pisarchik. Classification of external signal by spiking neural network of bistable hodgkin-huxley neurons. In *2020 4th Scientific School on Dynamics of Complex Networks and their Application in Intellectual Robotics (DCNAIR)*, pages 31–33. IEEE, 2020.

[92] Yanli Yao, Qiang Yu, Longbiao Wang, and Jianwu Dang. An integrated system for robust gender classification with convolutional restricted boltzmann machine and spiking neural network. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2348–2353. IEEE, 2019.

[93] Juan P. Dominguez-Morales, Qian Liu, Robert James, Daniel Gutierrez-Galan, Angel Jimenez-Fernandez, Simon Davidson, and Steve Furber. Deep spiking neural network model for time-variant signals classification: a real-time speech recognition approach. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[94] Nikola Kasabov and Elisa Capecci. Spiking neural network methodology for modelling, classification and understanding of eeg spatio-temporal data measuring cognitive processes. *Information Sciences*, 294:565–575, 2015.

[95] Konstantinos Demertzis and Lazaros Iliadis. A hybrid network anomaly and intrusion detection approach based on evolving spiking neural network classification. In *E-Democracy, Security, Privacy and Trust in a Digital World*, volume 441 of *Communications in Computer and Information Science*, pages 11–23. Springer International Publishing, Cham, 2014.

A OVERVIEW OF RELATED WORK IN SNN RESEARCH

Work	Task	Test accuracy (%)	Neuron model	Neural Code	Architecture	Learning Method
2021 [67]	XOR	100.0	LIF	Rate Coding	1 Layer FFNN[14]	Transfer from RNN
2018 [68]	Iris WBC Pima	98.7 99.3 78.6	NIDA[15]	-	NIDA	EA
2016 [69]	Iris WBC	97.0 96.4	LIF	Population Coding	1 Layer FFNN	EA[16]
2015 [70]	Appendicitis Haberman Heart Hepatitis Ionosphere Iris Liver	73.0 72.0 58.2 54.0 69.6 89.7 50.6	IF[17]	Population Coding	1 Layer FFNN	EA
2014 [71]	Stanford Liver Tumor Global Cancer Map Glioma Breast Cancer 11 Tumor Hepatocell	99.7 99.8 98.5 96.0 73.8 97.8	NLIF[18]	Rate Coding	2 Layer FFNN	Wavelet Mapping
2010 [72]	Iris WBC	95.3 96.7	LIF	Rate Coding	2 Layer FFNN	Merge of STDP, BCM[19]
2001 [73]	Iris	97.0	SRM	Population Coding	2 Layer FFNN	QuickProp

Table A.1: Comparing related works on tabular data and XOR classification using SNNs

[14] Feed Forward Neural Network (i.e. a fully connected feed-forward architecture).
[15] Neuroscience Inspired Dynamic Architecture. Neurons configured in 3D space with unlimited connectivity.
[16] Evolutionary Algorithm.
[17] No specific type of Integrate-And-Fire neuron was given in their work.
[18] Non-Linear Integrate-And-Fire neuron. The integration of the membrane potential is non-linear (e.g. exponential)
[19] Bienenstock-Cooper-Munro learning rule, which is a learning rule that approximates Hebbian Learning.

Work	Task	Test accuracy (%)	Neuron model	Neural Code	Architecture	Learning Method
2020 [74]	MNIST	∼90.0	SRM	Temporal Coding	1 Layer FFNN	Online STDP
2020 [75]	MNIST	95.0-96.0	Adaptive SRM	Rate Coding	Adaptive FFNN	STDP
	EMNIST (incl. letters)	70.0-80.0				
2020 [76]	MNIST	99.0	LIF	Rate Coding	6 Layer C-SNN[20] SpiNNaker Chip implementation	Transfer from CNN
		98.5				STBP[21]
2019 [77]	MNIST	89.2	LIF	Rate Coding	2 Layer FFNN	STDP
2019 [78]	Color feature classification	90.0	LIF	Population Coding	2 Layer FFNN	Tempotron
2018 [79]	CIFAR (2 classes)	98.5	LIF	Rate Coding	6 Layer C-SNN	STDP
	CIFAR (4 classes)	90.3				
	MNIST	96.3				
2018 [80]	MNIST	81.8	LIF	Rate Coding	1 Layer C-SNN	STDP
	Caltech	82.0				
	MNIST	80.1			1 Layer FFNN	
2015 [81]	Hoda	95.0	-	Rate Coding	2 Layer FFNN	STDP
2015 [82]	MNIST	91.5	LIF	Temporal Coding	1 Layer FFNN	STDP
		91.2				Transfer from ANN

Table A.2: Comparing related works on image data classification using SNNs

Work	Task	Test accuracy (%)	Neuron model	Neural Code	Architecture	Learning Method
2021 [67]	Wake Phrase Detection (DEMAND)	87.0	LIF	Rate Coding	1 Layer FFNN	Transfer from RNN
2021 [83]	N-MNIST	70.0	Probabilistic Neuron	-	2 Layer FFNN	Expectation Maximisation

[20] Convolutional SNN.
[21] Spatio-temporal backpropagation.

2021 [84]	Encrypted internet traffic (ISCX)	96.0	LIF	Rate Coding	2 Layer FFNN	Surrogate Gradient Learning
2021 [85]	Neural spike classification	96.0	LIF	NA	1 Layer FFNN	STDP
2021 [86]	Arousal Recognition (DEAP) Arousal Recognition (MAHNOB -HCI) Valence Recognition (DEAP) Valence Recognition (MAHNOB -HCI)	78.8 79.4 67.8 72.1	LIF	Temporal Coding	NeuCube	STDP
2021 [87]	MIT-BIH Arrhythmia (ECG)	91.0	IF	Rate Coding	1 Layer FFNN	Transfer from CNN
2020 [88]	Binary electric pulse classification	100.0	HH	NA	1 Layer FFNN	Barabasi-Albert
2020 [89]	Binary road type from mobile sensor data Anomaly Detection for unpaved roads Anomaly Detection for paved roads	99.9 100.0 99.8	-	Population Coding	2 Layer FFNN	Least Squares
2020 [76]	N-MNIST	98.2 97.9	LIF	NA NA	6 Layer C-SNN SpiNNaker chip implementation	Transfer from ANN STBP
2020 [90]	Motor Imagery Detection (EEG)	60.0-90.0 per subject 54.0-95.0 per subject	IZ[22]	NA NA	Single Neuron 2 Layer FFNN	Particle swarm Optimization
2020 [91]	Electric pulse Classification	100.0	HH	NA	1 Layer FFNN	Barabasi-Albert

2019 [92]	TIMIT (Gender)	98.0	LIF	Temporal Coding	1 Layer FFNN	Tempotron
2018 [68]	Radio (from DeepSig) Epilepsy Detection (EEG) TIMIT (vowel vs consonant)	71.0 99.0 85.0	NIDA	-	NIDA	EA
2018 [93]	Binary Speech Command Detection	89.9	LIF	NA	4 Layer C-SNN	STDP
2016 [69]	Epilepsy Detection (EEG)	89.3-91.1 per subject	LIF	Population Coding	1 Layer FFNN	EA
2015 [94]	Cognitive Process classification (EEG)	80.0-100.0 per subject	LIF	Temporal Coding	NeuCube[23]	STDP
2014 [95]	Network Anomaly Detection	97.7	-	Population Coding	2 Layer FFNN	EA
2010 [72]	TI46 subset (ASR[24])	95.3	LIF	Rate Coding	2 Layer FFNN	Merge of STDP, BCM

Table A.3: Comparing related works for time-series data classification using SNNs.

[22] Izhekevich neuron.
[23] NeuCube is a development environment for brain-like AI that uses spiking neurons as building blocks.
[24] Automatic Speech Recognition.

B SUPPLEMENTARY MATERIAL ABOUT MODEL DEVELOPMENT

Hyperparameter	**Grid values**
Δt	1e-2, 1e-3, 1e-4,
τ_{syn}	10e-4, 10e-3, 10e-2
τ_{mem}	10e-4, 10e-3,10e-2
Learning rate	1e-4, 1e-3, 1e-2
Batch size	128, 256, 512
Size of hidden layers	25, 50, 100, 200

Table B.1: Hyperparameter grid for greedy optimisation of the underlying SNN models.

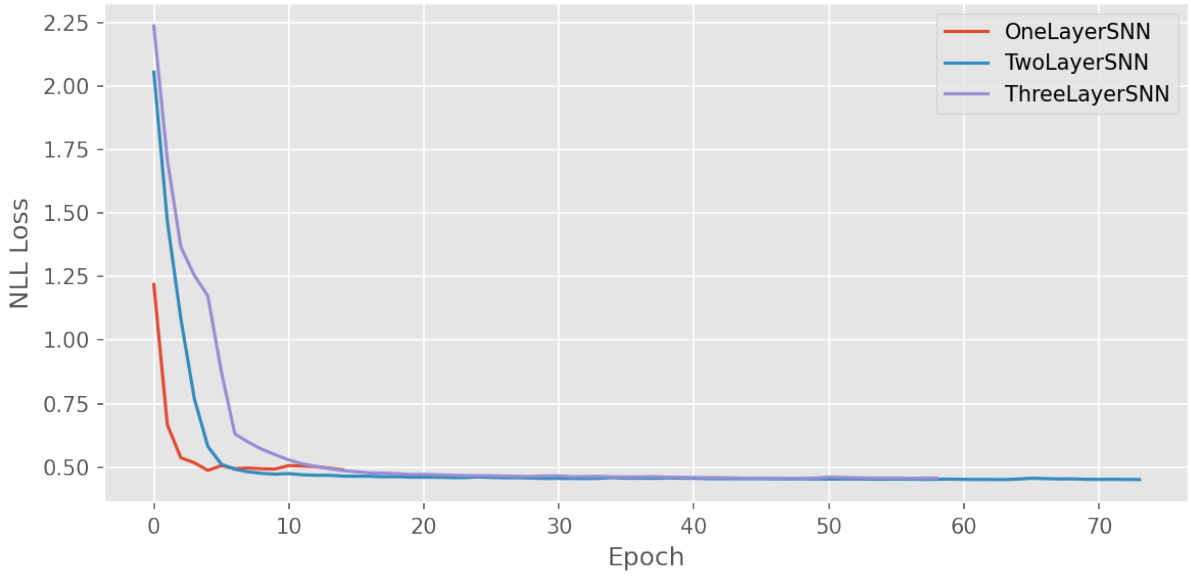
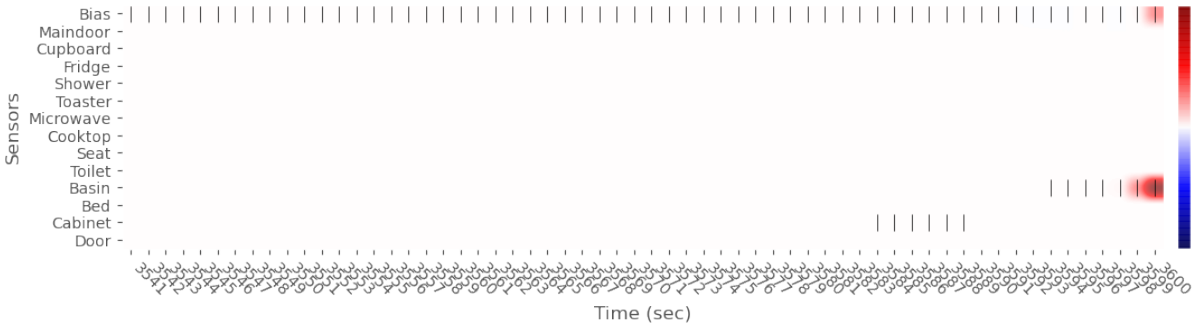


Figure B.1: Training loss history per SNN model trained with early stopping and a patience of 20 epochs.

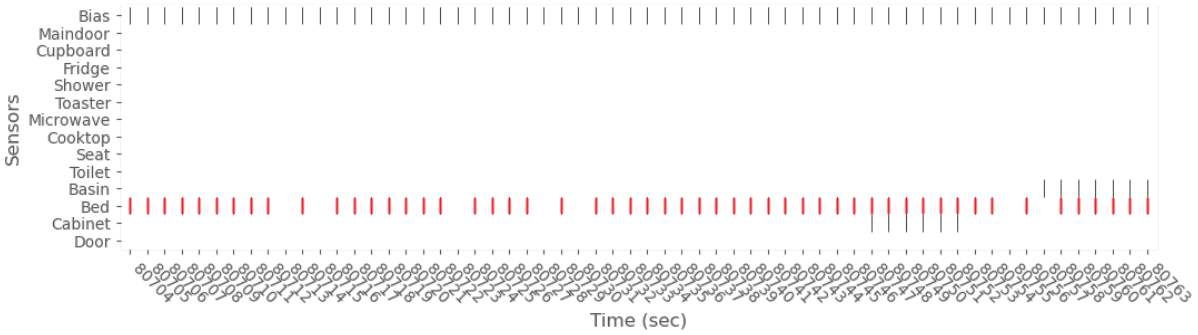
Hyperparameter	OneLayerSNN	TwoLayerSNN	ThreeLayerSNN
Δt			
0.01	1.594	1.606	2.398
0.001	**0.532**	**0.408**	**0.422**
0.0001	1.047	1.379	1.590
τ_{syn}			
0.1	0.748	0.826	0.805
0.01	**0.404**	**0.416**	**0.434**
0.001	0.753	0.419	0.462
τ_{mem}			
0.1	0.741	0.505	0.441
0.01	**0.418**	0.426	**0.431**
0.001	0.769	**0.404**	0.437
Learning rate			
0.01	**0.388**	0.456	0.479
0.001	0.507	**0.411**	**0.425**
0.0001	2.256	0.863	0.562
Batch size			
128	**0.389**	0.409	0.435
256	0.392	**0.405**	0.440
512	0.409	0.419	**0.427**
Hidden Layer size 1			
25	-	0.418	0.423
50	-	0.411	**0.416**
100	-	**0.406**	0.433
200	-	0.409	0.439
Hidden Layer size 2			
25	-	-	**0.420**
50	-	-	0.430
100	-	-	0.456
200	-	-	0.427

Table B.2: Validation negative log-likelihood loss of tuning process per SNN model. The validation loss is reported after a training of 20 epochs.

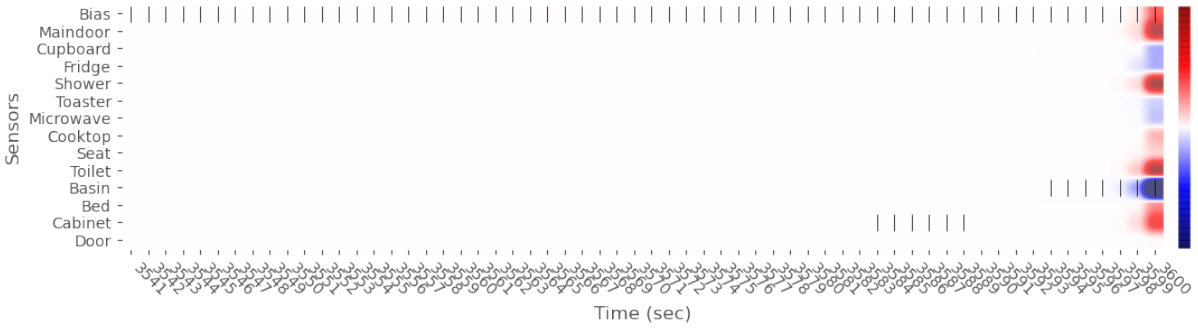
C EXAMPLES FROM THE QUANTITATIVE ANALYSIS



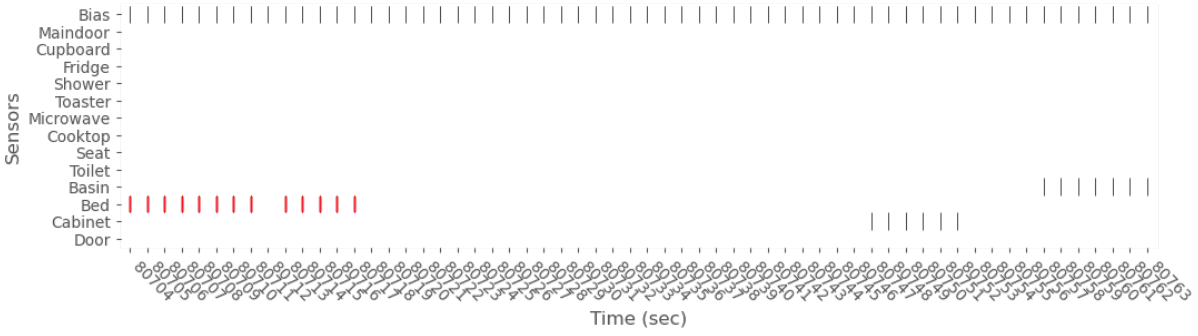
(a) TSA-S explanation for *TwoLayerSNN*'s prediction of timestep 80763.



(b) Time series until timestep 80763 with perturbed background based on the TSA-S explanation of *TwoLayerSNN*'s prediction.

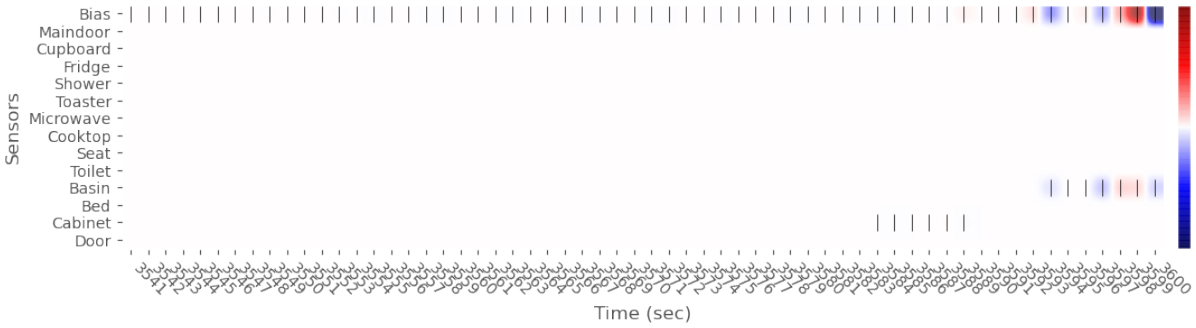


(c) TSA-NS explanation for *TwoLayerSNN*'s prediction of timestep 80763.

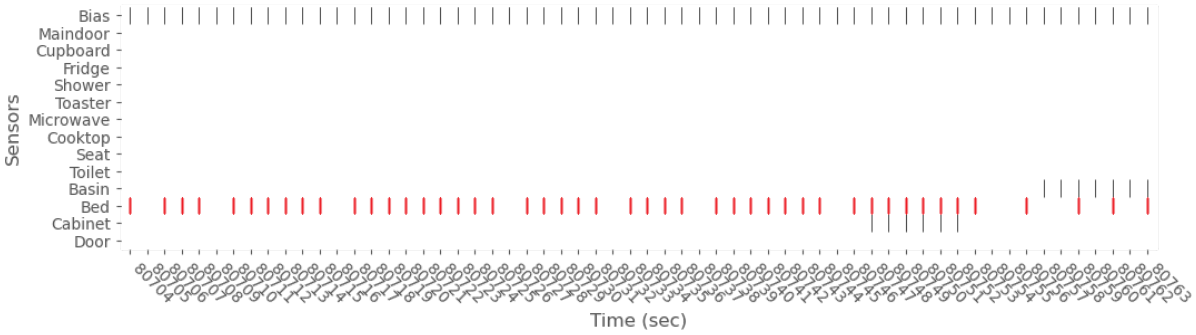


(d) Time series until timestep 80763 with perturbed background based on the TSA-NS explanation of *TwoLayerSNN*'s prediction.

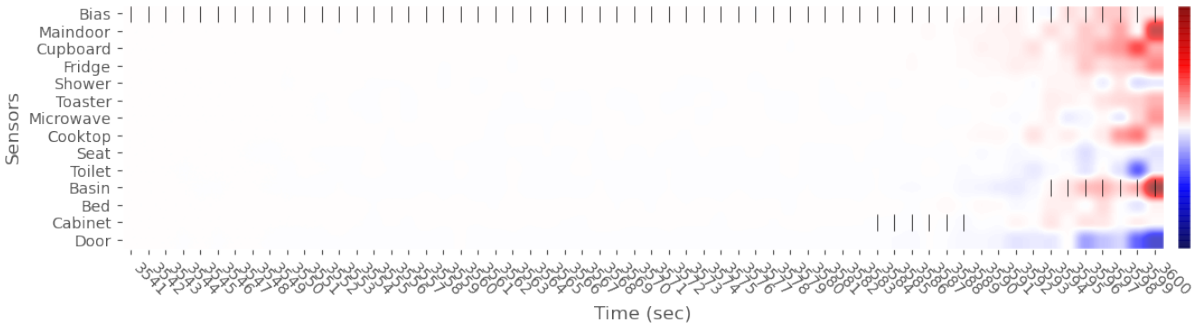
Figure C.1: Example of TSA-S and TSA-NS explanations of *TwoLayerSNN*'s prediction for timestep 80763 in (a) and (c). Red marks positive and blue negative attributions. (b) and (c) show the data with a shuffled background based on the explanations and $\theta = 0$ for the evaluation of attribution sufficiency. The red spikes show the difference between the perturbed and clean input. The perturbation based on TSA-NS does not reach the last steps before timestep 80763.



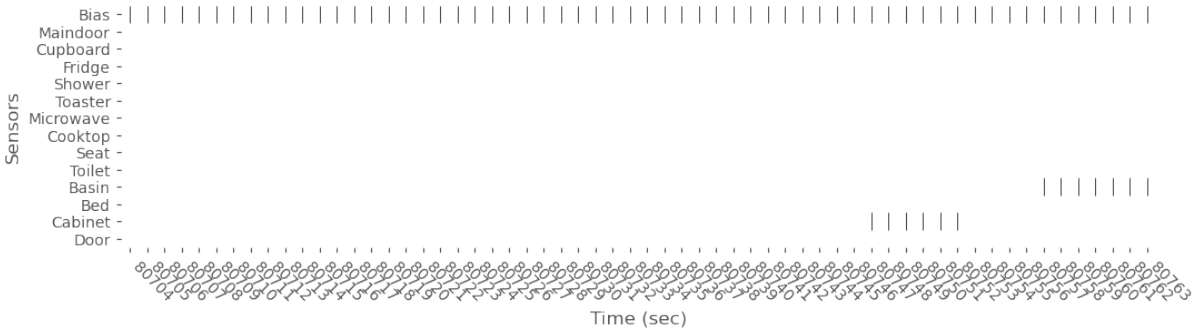
(a) TSA-S explanation for *ThreeLayerSNN*'s prediction of timestep 80763.



(b) Time series until timestep 80763 with perturbed background based on the TSA-S explanation of *ThreeLayerSNN*'s prediction.

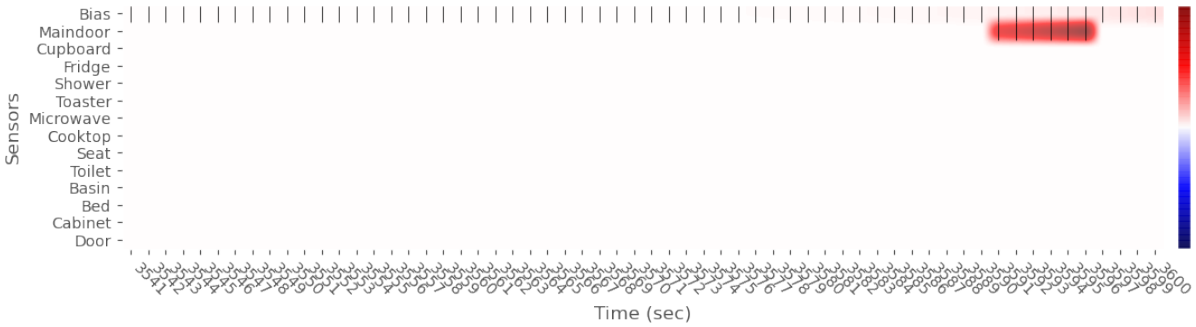


(c) TSA-NS explanation for *ThreeLayerSNN*'s prediction of timestep 80763.

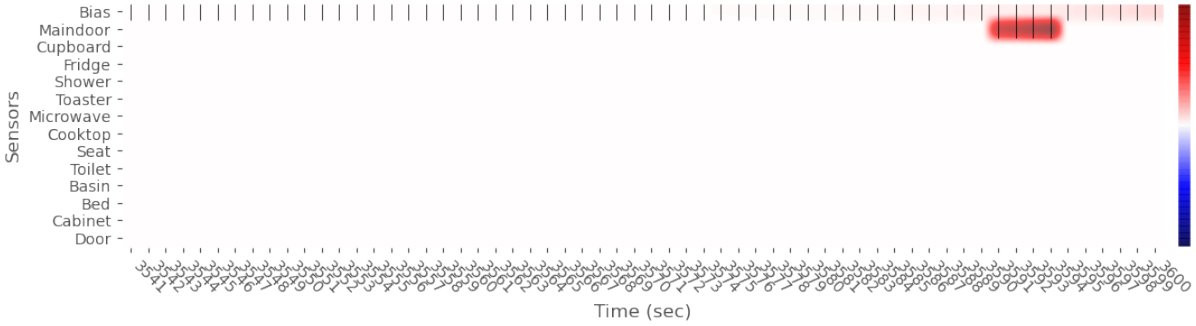


(d) Time series until timestep 80763 with perturbed background based on the TSA-NS explanation of *ThreeLayerSNN*'s prediction.

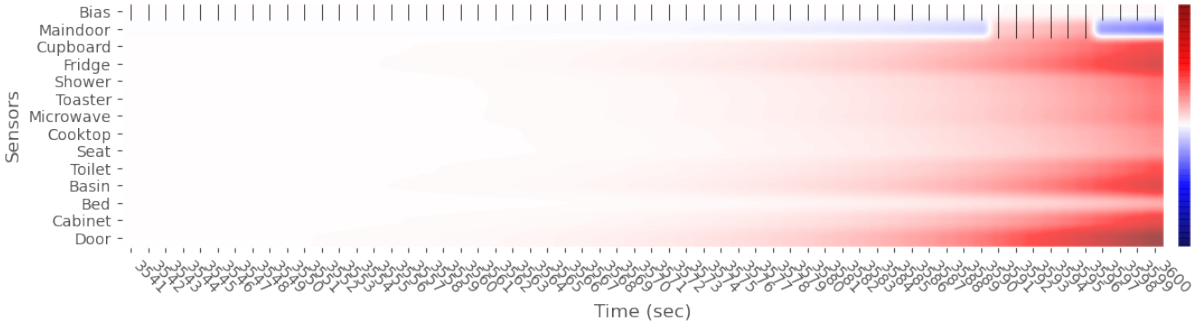
Figure C.2: Example of TSA-S and TSA-NS explanations of *ThreeLayerSNN*'s prediction for timestep 80763 in (a) and (c). Red marks positive and blue negative attributions. (b) and (c) show the data with a shuffled background based on the explanations and $\theta = 0$ for the evaluation of attribution sufficiency. The red spikes show the difference between the perturbed and clean input.



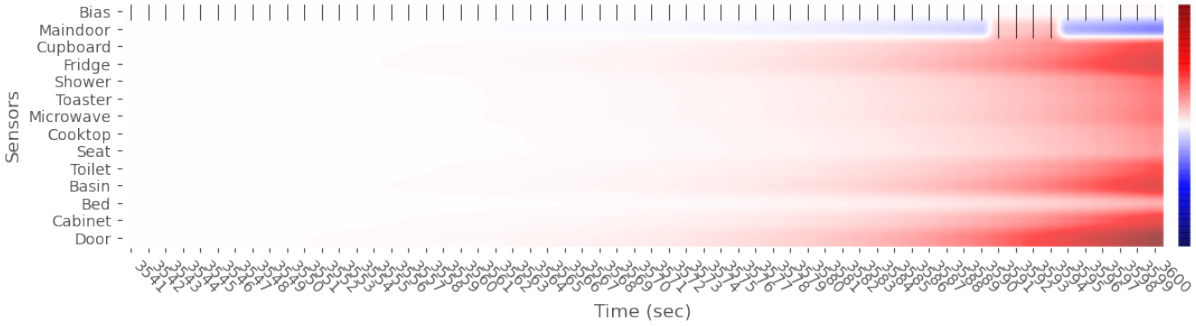
(a) TSA-S explanation for *OneLayerSNN*'s prediction of timestep 26442.



(b) TSA-S explanation for *OneLayerSNN*'s prediction of timestep 26442 with natural perturbation.

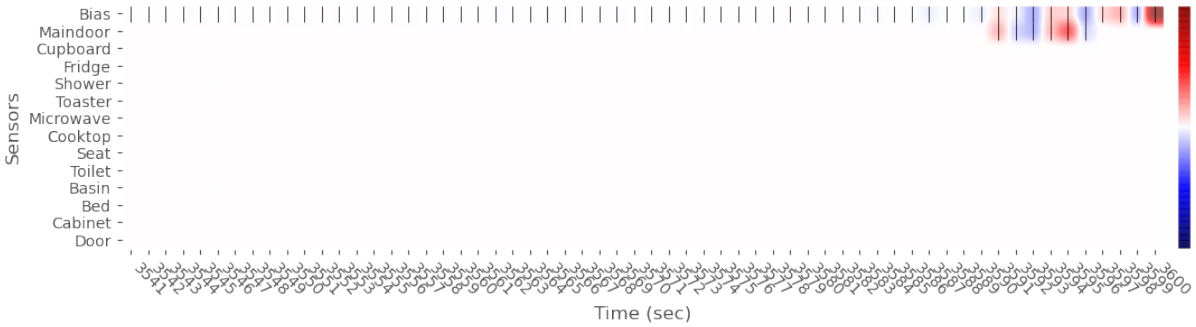


(c) TSA-NS explanation for *OneLayerSNN*'s prediction of timestep 26442.

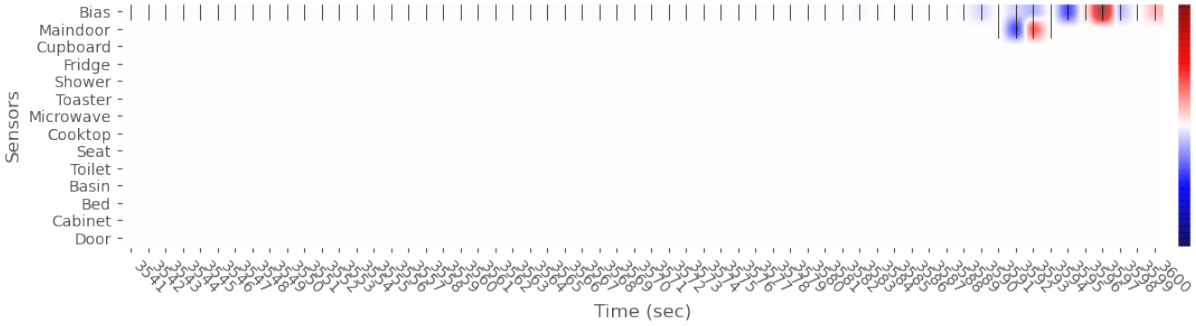


(d) TSA-NS explanation for *OneLayerSNN*'s prediction of timestep 26442 with natural perturbation.

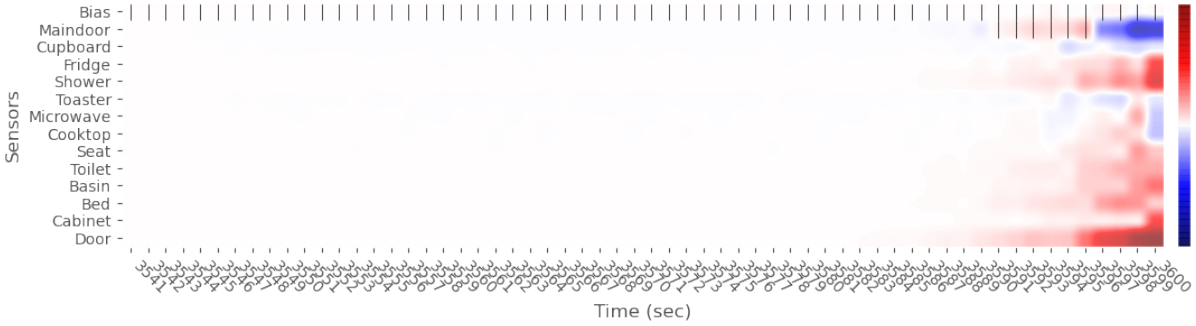
Figure C.3: Example of TSA-S and TSA-NS explanations of *OneLayerSNN*'s prediction for timestep 26442 with and without natural perturbation (shortening of the maindoor activation by two seconds) for the evaluation of stability. Red marks positive and blue negative attributions, where the colour corresponds to the attribution value. The explanation changes slightly for the perturbed example in both TSA variants.



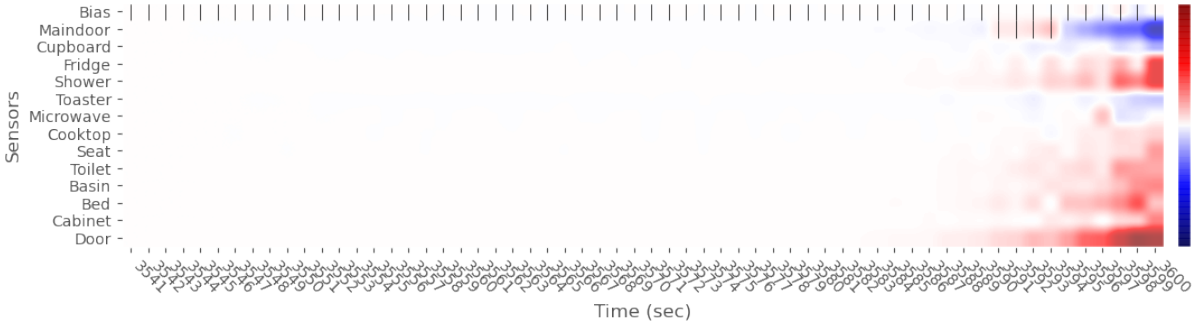
(a) TSA-S explanation for *ThreeLayerSNN*'s prediction of timestep 26442.



(b) TSA-S explanation for *ThreeLayerSNN*'s prediction of timestep 26442 with natural perturbation.



(c) TSA-NS explanation for *ThreeLayerSNN*'s prediction of timestep 26442.



(d) TSA-NS explanation for *ThreeLayerSNN*'s prediction of timestep 26442 with natural perturbation.

Figure C.4: Example of TSA-S and TSA-NS explanations of *ThreeLayerSNN*'s prediction for timestep 26442 with and without natural perturbation (shortening of the maindoor activation by two seconds) for the evaluation of stability. Red marks positive and blue negative attributions. The perturbation shows to impact the attributions of other features beside the perturbed features as well.

D UNMASKED SIMULATION EXPLANATIONS FOR USER STUDY

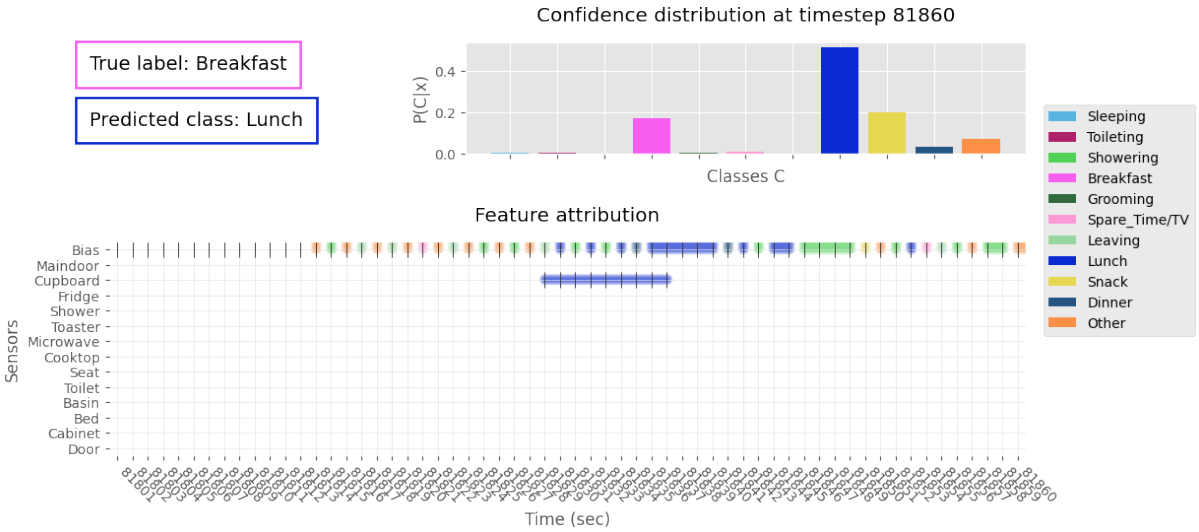


Figure D.1: Unmasked explanation for timestep 81860 of the testset

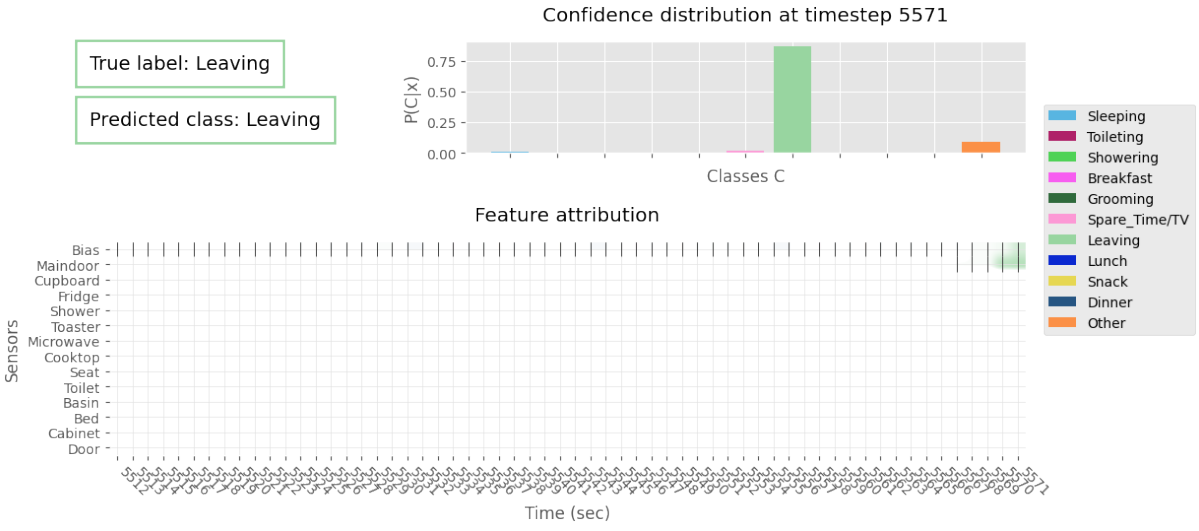


Figure D.2: Unmasked explanation for timestep 5571 of the testset

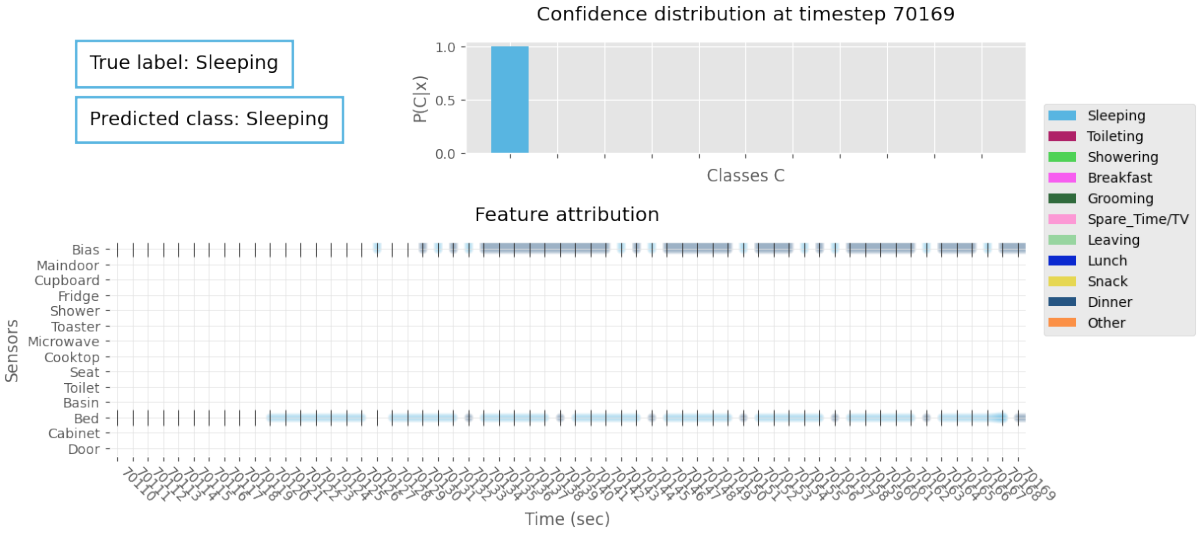


Figure D.3: Unmasked explanation for timestep 70169 of the testset

E CLUSTERING INSTRUCTIONS

First of all, thank you for willing to help me with the analysis of the survey responses to qualitatively evaluate my thesis.

In my thesis, I developed a feature attribution-based explanation for predictions from a spiking neural network model. As you have seen in the survey, this means that an explanation for why the model predicted a certain class is provided through highlighting which parts of the data attributed to which classes. This information shall help humans understand the model better and provide transparency. In other words, a bit of light shall shine into the black box that is a spiking neural network.

However, it is not guaranteed that this explanation is a good one. But what requirements does a good explanation have to fulfil? There are several aspects to consider besides the truthfulness of the explanation. Human-comprehensibility is one because the explanation is valuable to a human. This is assessed through the survey. Human-comprehensibility as a requirement details that an explanation should be clear and unambiguous, as well as understandable in terms of coherence with the human's background knowledge and beliefs.

In the survey, participants were presented with example explanations for example data and asked to interpret the explanations. How did they understand the model arrived at its prediction based on the provided explanation? This was a free-text answer question. In the next section of the mural, you will see all the answers to each of the three examples on different color sticky notes, one color corresponding to one example explanation.

Your task is to perform an inductive cluster analysis. This means that you should identify clusters of the user's interpretations, with one cluster corresponding to one possible interpretation. Please also describe the topic of the cluster. You are free to define as many clusters as you see fit. You can cluster the sticky notes together to form a cluster and add a description by adding some text.

In a nutshell, the instructions for this task are:

For each explanation:

1. Read the survey responses.
2. Identify themes of people's interpretations that you recognise (no set number, up to you).
3. Note these on WHITE sticky notes (can be found under 'Text' on the left side of the Mural).
4. Group the survey response sticky notes under these themes.
5. Add any comments on other WHITE sticky notes

Are you ready? Let's go to the next section!

F RESULTS OF INDUCTIVE CLUSTER ANALYSIS

Explanation	Original clusters A	Count	Umbrella cluster
#1	Both features and probability	11	Data and confidence
	Only feature attribution	15	Data
	Only probability distribution	3	Classification confidence
	Uncertain/unclear	3	No clear answer
	Personal interpretation	1	Learned patterns
#2	Only feature attribution	15	Data
	Only probability distribution	2	Classification confidence
	Both features and probability	13	Data and confidence
	Personal interpretation	1	Learned patterns
	Used time of day as the only feature?	1	Learned patterns
#3	Only feature attribution	24	Data
	Both features and probability	7	Data and confidence
	Only probability distribution	2	Classification confidence

Table F.1: Clusters defined by Annotator A per explanation with mapping to the umbrella clusters.

Explanation	Original clusters B	Count	Umbrella cluster
#1	Many explanations: sensor activation, confidence, bias, time	4	Data and confidence
	Sensor activation, time, seat, basin	19	Data
	confidence	4	Classification confidence
	Not clear / unsure	3	No clear answer
	Learned patterns during training	2	Learned patterns
	Bias	1	Learned patterns
#2	Opening the cupboard / use of cupboard	7	Data
	Use of cupboard and model training	1	Learned patterns
	colour(s) in the figures/plots	1	Data and confidence
	Sensor activation	8	Data
	confidence	3	Classification confidence
	sensor activation and confidence	7	Data and confidence
	Time and sensor activation	1	Data
	Using cupboard and time	2	Data
	Model training	1	Learned patterns
	Time	1	Learned patterns
#3	Using seat	6	Data
	Sensor activation and time	4	Data
	Sensor activation	13	Data
	Color(s) in the figures	1	Data and confidence
	Using seat and time	1	Data
	Sensor activation and confidence	4	Data and confidence
	Sensor activation, confidence and time	1	Data and confidence
	Confidence	3	Classification confidence

Table F.2: Clusters defined by Annotator B per explanation with mapping to umbrella clusters.

Explanation	Original clusters C	Count	Umbrella cluster
#1	Seat used for long time (longer than basin)	20	Data
	Unsure due to confidence distribution (Does not understand the ”other” category); Seat used for long time (longer than basin)	2	Data
	Participant unsure/does not understand; When seat is used it is classified as spare time	1	Data
	Participant unsure/does not understand; Seat used for long time (longer than basin)	1	Data
	When seat is used it is classified as spare time	2	Data
	Highest classification confidence	4	Classification confidence
	Participant unsure/does not understand	1	No clear answer

	Use of toilet during spare time was still classified as spare time in training data	1	Data
	Participant misunderstood the meaning of true label/predicted class	1	No clear answer
#2	Cupboard sensor activation indicates lunch	12	Data
	Unsure due to confidence distribution (Confidence distribution is spread out between classes); Cupboard sensor activation indicates lunch	2	Data
	Color (Because model says so)	1	Data
	Lunch has the highest confidence	3	
	Lunch has the highest confidence; Unsure due to confidence distribution(Confidence distribution is spread out between classes); Cupboard sensor activation indicates lunch	2	Data
	Lunch has the highest confidence; Cupboard sensor activation indicates lunch	3	Data and confidence
	Lunch has the highest confidence; Model biased to choose lunch; Cupboard sensor activation indicates lunch	1	Learned patterns
	Unsure due to other kitchen sensors being inactive (Kitchen appliances are not used); Model considers the time of day; Cupboard sensor activation indicates lunch	1	Learned patterns
	Unsure as other kitchen sensors are being inactive (Kitchen appliances are not used); Model biased to choose lunch; Cupboard sensor activation indicates lunch	1	Learned patterns
	Model considers the time of day; Cupboard sensor activation indicates lunch	2	Learned patterns
	Model biased to choose lunch	1	Learned patterns
	Model biased to choose lunch; Cupboard sensor activation indicates lunch	2	Learned patterns
	Model considers the time of day	1	Learned patterns
#3	Seat is used	19	Data
	No other sensors activated; Seat is used	5	Data
	Color (Because model says so)	1	Data
	High confidence; No other sensors activated; Seat is used	3	Data and confidence
	High confidence; Seat is used	3	Data and confidence
	High confidence	2	Classification confidence

Table F.3: Clusters defined by Annotator C per explanation with mapping to umbrella clusters

ATTACHMENTS

Attached to this thesis are the following documents:

1. Ethics approval for the user study from the Ethics Committee Computer & Information Science at the University of Twente,
2. Survey design of the user study,
3. Original coding of survey answers,
4. Calculation of Fleiss' kappa for the code of the user study.

EC-CIS **ADVICE FORM**

RP 2021-195	Local feature-based explanations for spiking neural networks
Name of reviewer	Dennis Reidsma
Date	20 sept 2021

Conflict of interest of the reviewer
For the reviewer: please indicate whether you have an interest in the research and if relevant, describe the closeness, severity and consequences

none

Review

- The proposal is not only well prepared but also excellently documented; everything is very clear. THANK YOU.
- No concerns; positive advice

X No adjustments needed. Secretary can send advice.

☐ Needs minor adjustments, please, send to ethicscommittee-cis@utwente.nl. Secretary can send advice after receiving adjustments.

☐ Needs major adjustments, please send to me for review, with cc to ethicscommittee@utwente.nl.

Comprehensibility of explanations from spiking neural networks

Thank you for participating in my survey about the quality of explanations from spiking neural networks, which is part of my Master thesis research at the University of Twente.

Purpose:
By participating in this research, you will help me evaluate my research and find out whether my explanations for the predictions of a spiking neural network are human-understandable. Your responses are intended for qualitative evaluation of my method and is not intended to test your performance in any way.

Background:
In my Master thesis, I studied local explanations (i.e., explaining single predictions) based on features (i.e., by looking at the input features) from spiking neural networks (SNNs) using an activity of daily living prediction task (i.e., predicting what a person is doing based on sensor activity). SNNs are known as the third generation of neural networks and are biologically more plausible than neural networks based on artificial neurons. They are not popular because the research community has not found an efficient learning method for them yet. However, they are in theory at least as powerful as their predecessors and predestined to be used in critical application areas, e.g., health or traffic, which is why I am studying how to generate a model-specific explanation method for these types of networks and shine some light into the black box.

Procedure:
First, a general question will be asked and some more background information about the data and task given. Then, you will be presented with example explanations and asked to give your interpretation and understanding of those. In the last part, you will be asked to classify a small data sample based on the data and the explanation. This survey will take approximately 15-30 minutes. It is recommended to fill this survey on a laptop.

Anonymity:
I will use your answers, as well as the answers from other respondents to qualitatively evaluate my research. The answers are recorded anonymously, and therefore there is no connection from your answers to your person.

Freedom to withdraw:
You may withdraw from the activity at any time without penalty, by not submitting the survey.
If you have any questions, feel free to contact me at t.q.e.nguyen@student.utwente.nl.
If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher, please contact the Secretary of the Ethics Committee "Computer & Information Sciences" of the University of Twente, drs. Petri de Willigen, mail: ethicscommittee-cis@utwente.nl.

*Required

1. Active consent *

Tick all that apply.

- [ ] I have read and understood the above information.
- [ ] I agree that the results of this study can be used for academic purposes.

Inclusion criteria

The target group for the explanation method are model developers. Therefore, only participants with some form of familiarity with supervised machine learning are in my target group for this survey.

2. Do you have some prior knowledge and/or experience in the field of supervised machine learning? *

Mark only one oval.

- ◯ Yes
- ◯ No *Skip to section 13 (Thank you.)*

Preliminary information

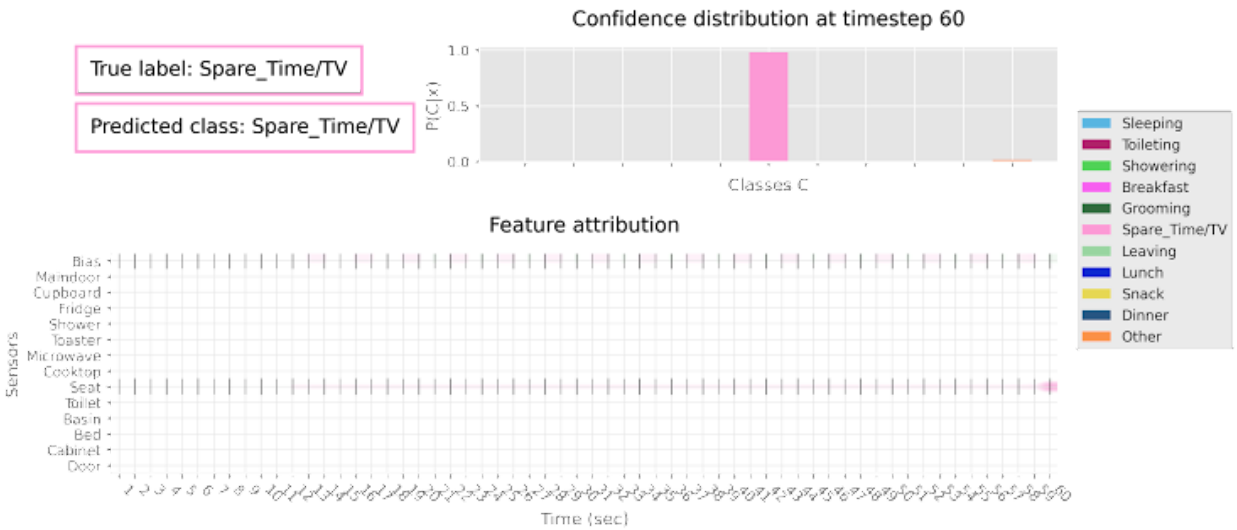
In this survey, you will help me evaluate an explanation for a machine learning model's prediction on certain input data. In this short information part, you will be briefed on the data and task.

What kind of data is used?
For the research, a model was trained on the Activities of Daily Living from Binary Sensors dataset from the UCI ML repository [1]. It is a multivariate time series dataset of sensor data. The sensors are placed around a home, for example on the bed or on the toilet. Since they are binary sensors, they are only either activated or not. For a spiking neural network, the sensor activation has been coded into spikes, where spiking indicates that the sensor is activated. Each sensor can spike at most once per timestep. A constantly spiking bias sensor was added in addition to the sensors available in the dataset (similar to a bias neuron with value 1 in a deep neural network). The sensors can be seen on the y-axis of the graph below.

What is the task?
The model was trained to predict the activity of the person living in the home based on the sensor data, e.g. 'eating' or 'leaving'. The model predicts this continuously at each time step. The dataset is originally annotated with 10 classes. Some parts were not annotated, and put into an "Other" class. The classes can be seen in the legend of the graph below.

[1] Javier Ordóñez, Paulade Toledo, and Araceli Sanchis. Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. Sensors(Basel, Switzerland), 13(5):5460–5477, 2013.

3. Example explanation: Do you understand the components (true label, predicted class, confidence distribution, feature attribution) of the explanation? *



High resolution image available here: https://drive.google.com/file/d/1-UCKtiCf-rZCfELXx8kyENLBnrzzL7cn/view?usp=sharing

The image above shows an example explanation of a SNN model for a data sample from timestep 1 to timestep 60, during which the seat sensor and bias sensor are spiking at each timestep. There are 3 parts to the explanation:
(1) The predicted and true label at the top left corner.
(2) The classification confidence of the current time step on the top right corner.
(3) The feature attribution highlights on the bottom part overlayed with the data.

In this example, the model correctly predicts that the person is watching TV in their spare time, and the seat sensor activation from timestep 60 mainly contributes to this prediction at timestep 60.

Do you understand the components of the explanation?

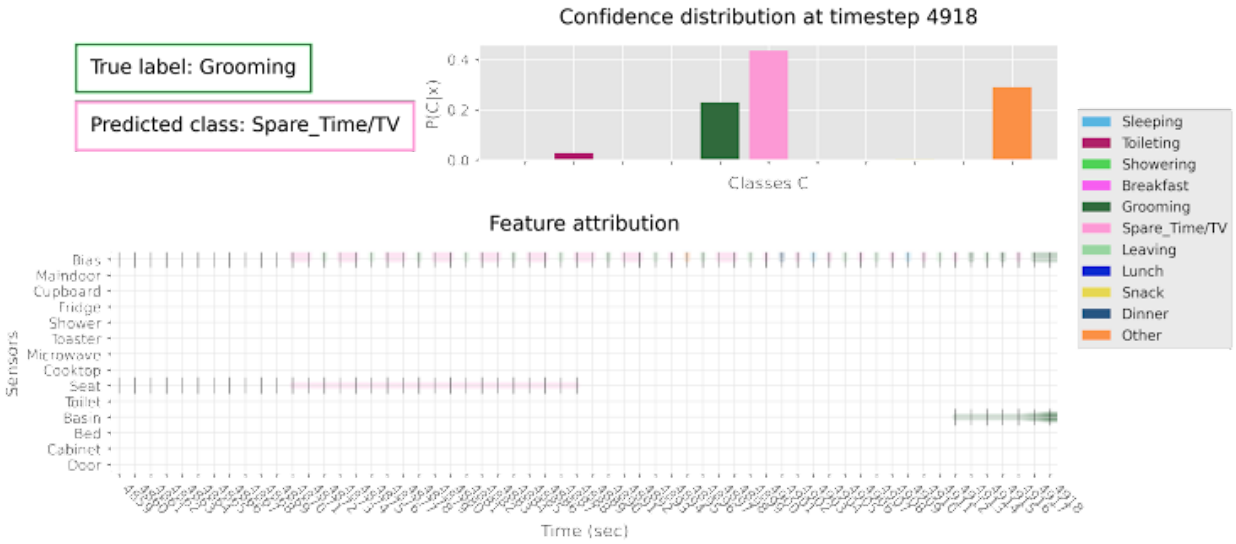
Tick all that apply.

- [ ] Yes
- [ ] No
- [ ] Not sure (Please detail what is unclear in "Other")
- [ ] Other: _____

Explanation understanding

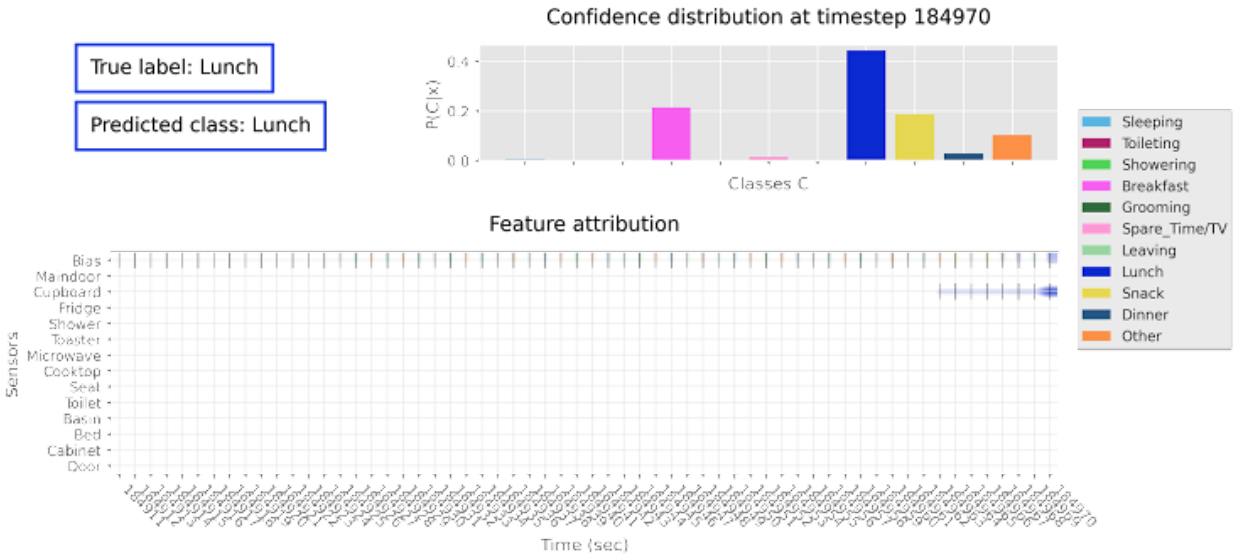
In this part of the survey, you will be shown 3 data samples with explanations for the model's predictions. Please take a look at the explanation, and write a short paragraph of how you understand it.

4. In the example below, why did the model predict >Spare_Time/TV<? *



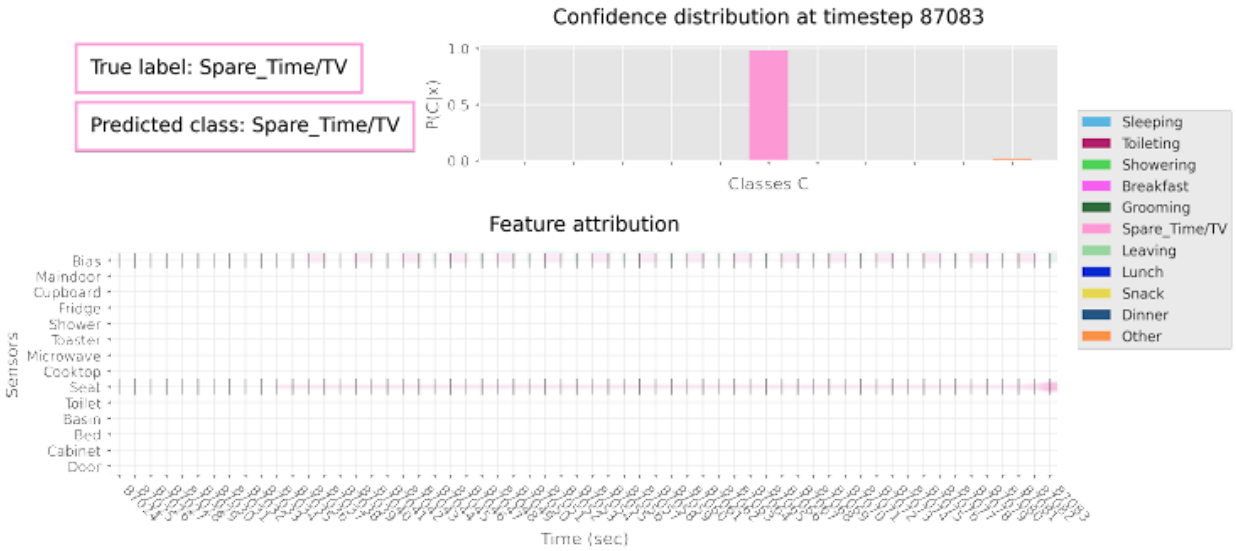
High resolution image here: https://drive.google.com/file/d/1-UwEJ8oPh6dgvtcYKBe0oPf3Y_daHae-/view?usp=sharing

5. In the example below, why did the model predict >Lunch<? *



High resolution image here: https://drive.google.com/file/d/1-VhevBMKxs0CKYiBTF1JwL4lK5hQvwho/view?usp=sharing

6. In the example below, why did the model predict >Spare_Time/TV<? *



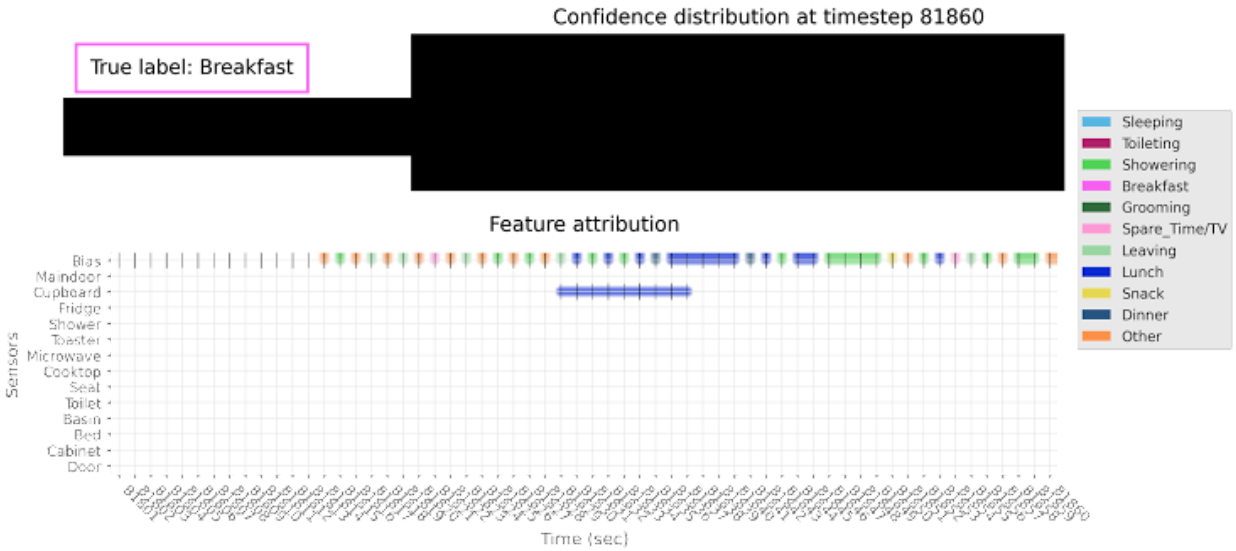
High resolution image here: https://drive.google.com/file/d/1-aTdS34itjdlIlP6--_rLgn3PjkHI5SF/view?usp=sharing

Simulation

In this part of the survey, you will be shown three data sample with the explanations, but without the model predictions and the confidence distributions (covered in black). Your task is to imitate the model and simulate its behaviour based on the explanations given.

Simulation - Part 1/3

7. In the example below, what would be the model's prediction in your opinion? *



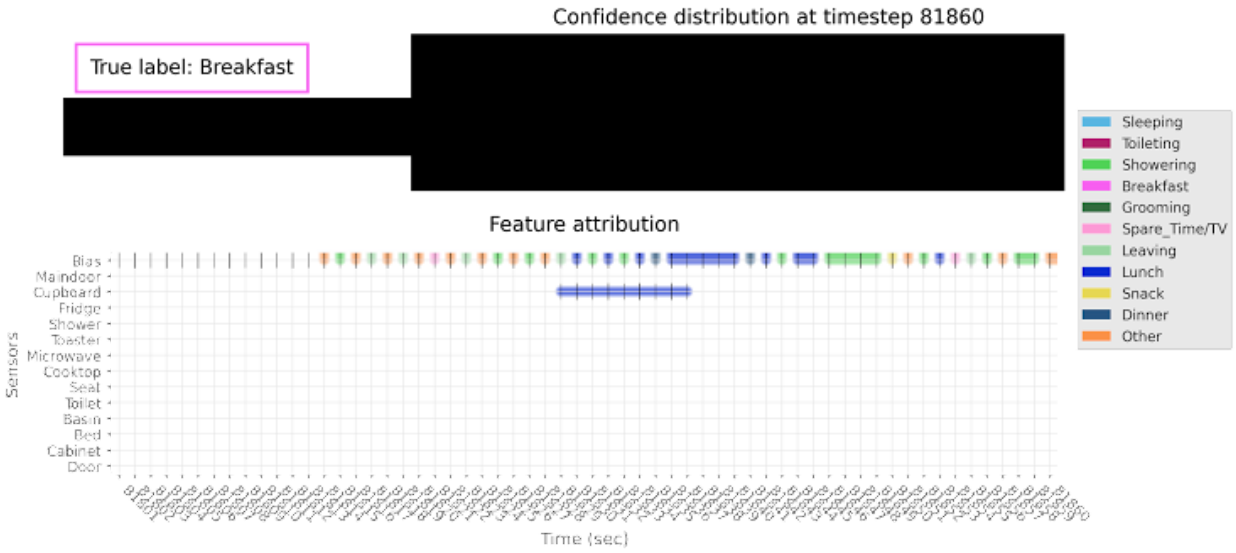
High resolution image here: https://drive.google.com/file/d/1-cJnkQzoWOe3o2f6Nql2hWDuzcHQHcgw/view?usp=sharing

Mark only one oval.

- ◯ Sleeping
- ◯ Toileting
- ◯ Showering
- ◯ Breakfast
- ◯ Grooming
- ◯ Spare_Time/TV
- ◯ Leaving
- ◯ Lunch
- ◯ Snack
- ◯ Dinner
- ◯ Other class
- ◯ I don't know

Simulation - Part 1/3

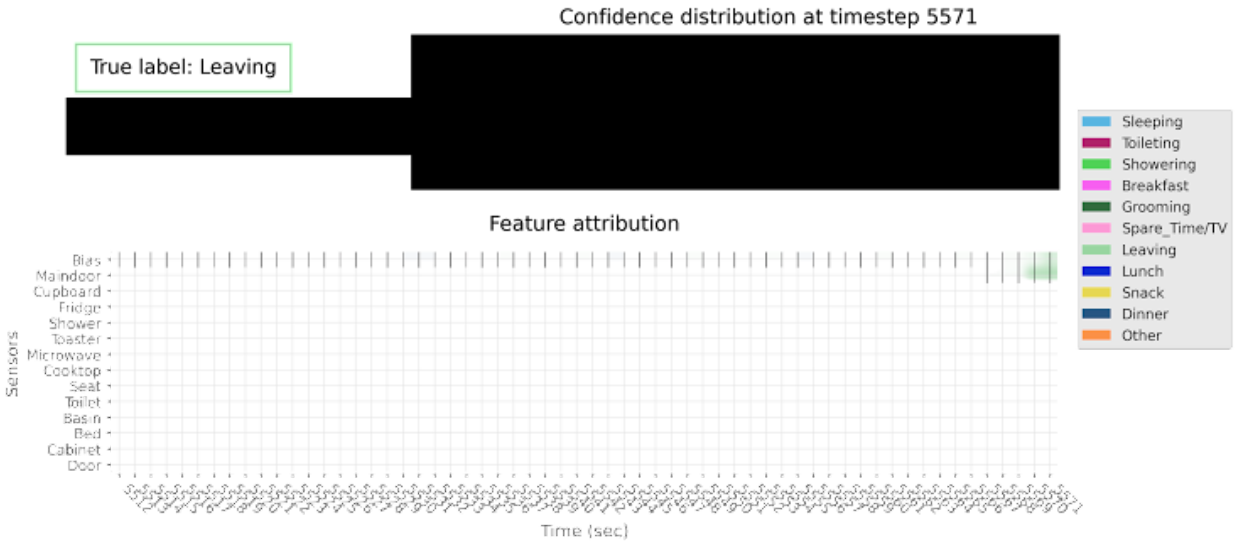
8. Please explain your prediction for the model's behaviour at second 81860. *



High resolution image here (same as before): https://drive.google.com/file/d/1-cJnkQzoWOe3o2f6Nql2hWDuzcHQHcgw/view?usp=sharing

Simulation - Part 2/3

9. In the example below, what would be the model's prediction in your opinion? *



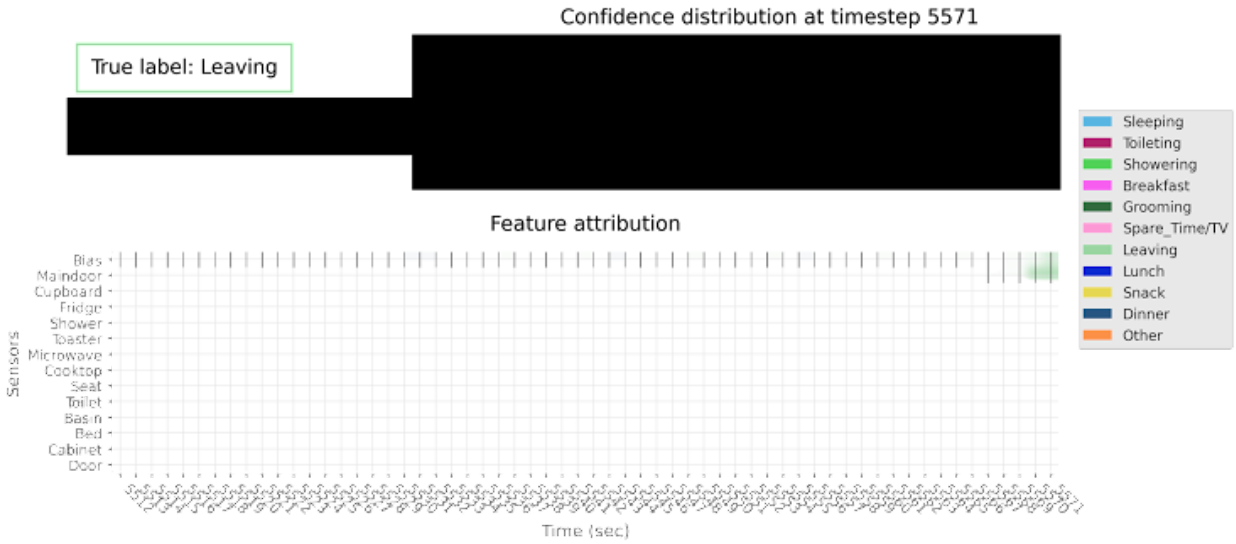
High resolution image here: https://drive.google.com/file/d/1-cT3GpY80yhWV2eNOgJ82_pkFwezRzqk/view?usp=sharing

Mark only one oval.

- Sleeping
- Toileting
- Showering
- Breakfast
- Grooming
- Spare_Time/TV
- Leaving
- Lunch
- Snack
- Dinner
- Other class
- I don't know

Simulation 2/3

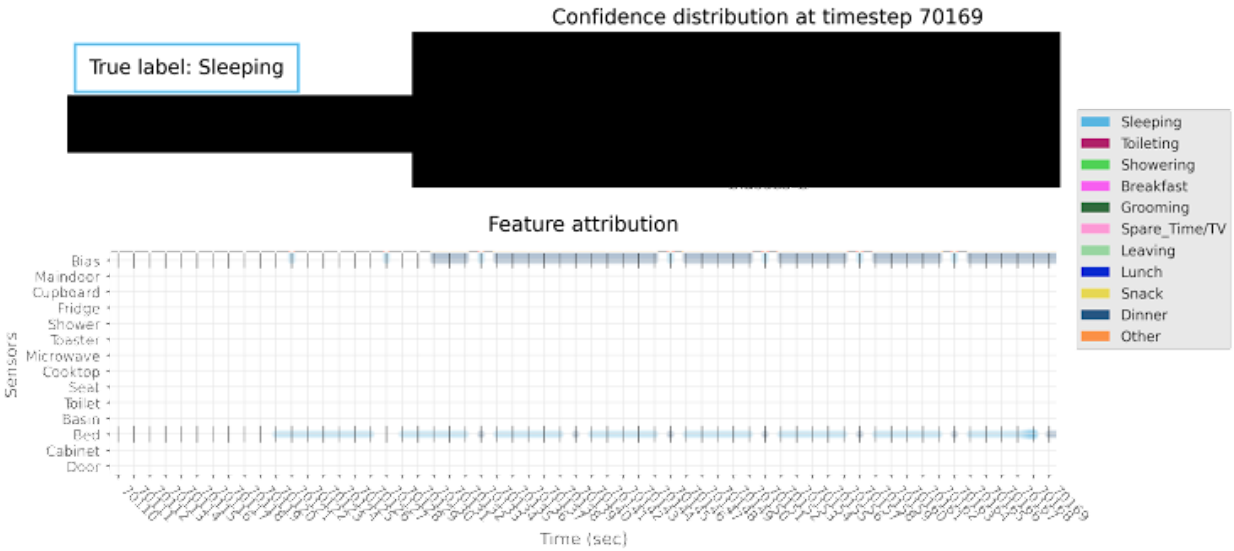
10. Please explain your prediction for the model's behaviour at second 5571. *



High resolution image here (same as before): https://drive.google.com/file/d/1-cT3GpY80yhWV2eNOgJ82_pkFwezRzqk/view?usp=sharing

Simulation - Part 3/3

11. In the example below, what would be the model's prediction in your opinion? *



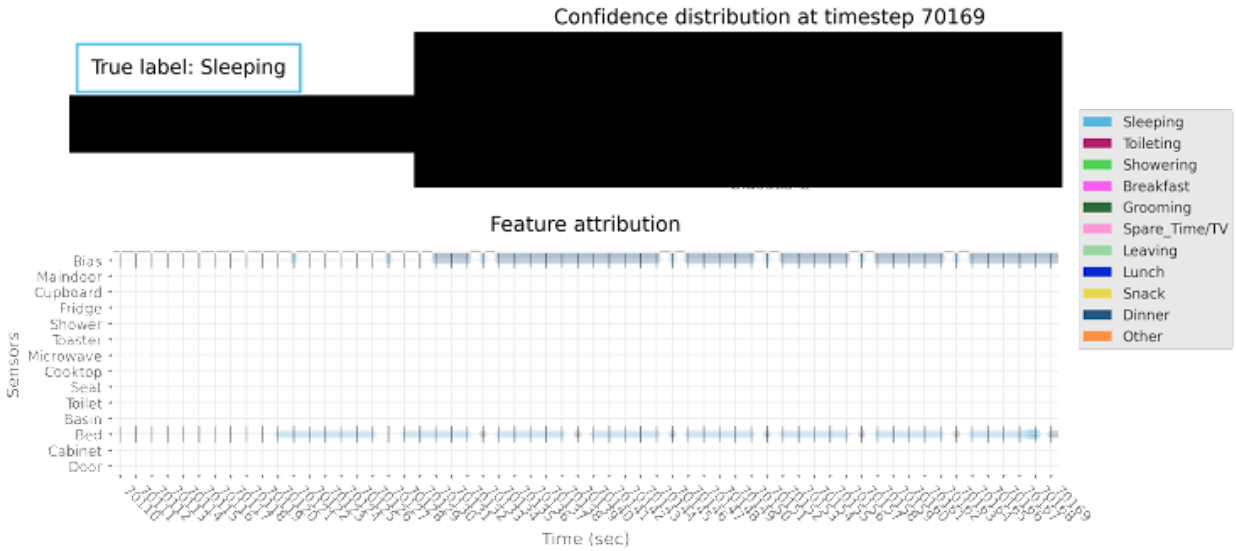
High resolution image here: https://drive.google.com/file/d/1-fuodqFrjej-u5ZixIorVeGdZNsmWwg9/view?usp=sharing

Mark only one oval.

- Sleeping
- Toileting
- Showering
- Breakfast
- Grooming
- Spare_Time/TV
- Leaving
- Lunch
- Snack
- Dinner
- Other class
- I don't know

Simulation 3/3

12. Please explain your prediction for the model's behaviour at second 70169. *



High resolution image here (same as before): https://drive.google.com/file/d/1-fuodqFrjej-u5ZixIorVeGdZNsmWwg9/view?usp=sharing

Thank you for your answer.

13. If you have any comments before submitting, please enter them below.

Thank you.

Thank you for your participation. Unfortunately, you are not within the target group of my research and therefore cannot submit a response. No data has been saved, and you can safely close the survey.

If you want to know more about the research or have other questions, feel free to contact me at t.q.e.nguyen@student.utwente.nl. I will be happy to answer any questions.

Coding of survey answers

Survey responses	Coding A	Umbrella cluster A	Coding B	Umbrella Cluster B	Coding C	Umbrella Cluster C	Comment
Because the seat was activated longer than the bassin and the confidence is nearly twice as high for tv than for grooming	Both features and probability	Data and confidence	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
Because the first half of the time was spent on the seat	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
The big distribution of class spare_time is influenced by the long spanning and more bias of the feature seat, which makes this class distribution have the highest value and thus the model chose this class. On the other hand, another active sensor was spiked at basin sensor (which indicates a stronger detection) contributes to class grooming to be activated. However, the confidence distribution of class grooming is not as high as the spare_time class so the model could not choose it. This can indicate that seat feature is seen as some sort of noises and together with a big amount of noises, the result will be influenced	Both features and probability	Data and confidence	Many explanations: sensor activation, confidence, bias, time	Data and confidence	Seat used for long time (longer than basin)	Data	
Similar to the previous answers, I would once again base this on the highest classification confidence.	Only probability distribution	Classification confidence	confidence	Classification confidenc	Highest classification confidence	Classification Confidence	
Because the person was sitting on the seat first, which lasted longer in this timeframe than going the the basin.	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
He/she spent most of the time on seat according to sensor, but then it appears that he/she decided to leave and basin sensor is activated at timestep 4918. So it should be grooming.	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
The seat was active with the most spikes and therefore the model predicts that. Still, the confidence is only 0.4 of the model. However, the true label was only the third largest indicator, probably because it was only briefly active. Not sure about the 'Other' category though.	Both features and probability	Data and confidence	Many explanations: sensor activation, confidence, bias, time	Data and confidence	Unsure due to confidence distribution (Does not understand the "other" category); Seat	Data	
the confidence is higher for Spare_Time then Grooming for timestep 4918	Only probability	Classification confidence	confidence	Classification confidenc	Highest classification	Classification	
perhaps, the duration at the seat is longer than at the toilet. So it predicted with higher confidence to Spare time class	Both features and probability	Data and confidence	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
detection at seat chich relates to spare time appears more, and in this case has a higher probability compared to the basin, which corresponds to grooming.	Both features and probability	Data and confidence	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
Until second 4877, the seat sensor is spiking. After that, no sensors record any spiking until second 4912. From there until the end, the basin sensor records an action. The model predicts four classes, with the class "spare_time/TV" having the highest prediction probability of a bit more than 0.4. The true label is grooming. The label was most likely predicted because the seating action is active longer than the basin action and seating probably relates to the spare_time/TV class.	Both features and probability	Data and confidence	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
First the seat sensor was activated and shortly after that, up until the time of interest (4918) the basin was activated. The model learned during training that this is most probably related to the activity "Grooming". However in this case the prediction is wron as it differs from the true label.	Only feature attribution	Data	Learned patterns during training	Learned patterns	Participant misunderstood the meaning of true label/predicted class	No clear answer	
In this window the largest part of the time the seat is activated, so overall the most plausible action would be watching TV. At the last few time steps the activation is indicating that the basin is activated which lead to grooming, which would be the most logical at that current time step. As seen the confidence of grooming is lower than the spare time/TV, probably due to the fact that the latter has a longer duration.	Both features and probability	Data and confidence	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
For the first part of the time series, the seat sensor is activated along with the bias sensor. This leads the model to predict the activity as Spare_Time/TV. However, towards the end the basic sensor is activated, while the seat sensor is not. However for the given time series, the feature attribution graph shows that the higher confidence for Spare_Time/TV could be attributed to the longer duration that the seat sensor was activated, which usually corresponds to Spare_Time/TV as seen from previous examples. It is unclear why the confidence of Other is so high when the feature attribution graph does not clearly show this.	Both features and probability	Data and confidence	Many explanations: sensor activation, confidence, bias, time	Data and confidence	Unsure due to confidence distribution (Does not understand "other" category); Seat used for long time (longer than basin)	Data	
I think that the bottom might be the the sensor data throughout the time now. So that we can see that grooming is happening at 4918. But the upperpart shows the predictions and shows that spare_time/tv is predicted since it has the highest p(C|x)	Both features and probability	Data and confidence	Sensor activation, time, seat, basin	Data	Highest classification confidence	Classification Confidence	
It is not clear to me wether the prediction is about the final time step or the whole time interval. If it is about the whole time interval it predicts spare_time/TV because that is predictablr for most of the time. If it is only for the final time step, the SNN is slow in response to a new input and therefore still puts spare_time/tv on top while for the later period the basin sensor was active.	Uncertain/unclear	No clear answer	Not clear / unsure	No clear answer	Participant unsure/does not understand; Seat used for long time (longer than basin)	Data	
Because the majority of the time, the person was sitting which seems to indicate spare time TV.	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
Because the seat was used	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	When seat is used it is classified as spare time	Data	
More seat activation then Basin.	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	

At timestep 4918 only bias and basin sensors were activated which would normally result in the prediction "Grooming". Nevertheless, the model predicted "Spare_Time/TV" because the seat sensor was activated longer than the basin sensor but shortly before its activation. As a result the probability for the label "Spare_Time/TV" to be true is still higher than for "Grooming".	Both features and probability	Data and confidence	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
Most time was spend on the seat.	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
Because the seat sensor was firing/spiking. Looks like they got up to pee so P wasn't 1?	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
Not clear to me	Uncertain/unclear	No clear answer	Not clear / unsure	No clear answer	Participant unsure/does	No clear answer	
same as before, it looks like there is a direct connection between the seat sensor and sparetime. Model bias I guess.	Only feature attribution	Data	Bias	Learned patterns	When the seat is used it is classified as spare time	Data	
the duration for activated sensor of seat is longer than the duration of basin in the past?	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
similar to above answer, in step 4918, we can observe a spike, in which, the Spare_Time class has the highest confidence among the others	Only probability distribution	Classification confidence	confidence	Classification confidenc	Highest classification confidence	Classification Confidence	
In this case, there are two different sensors activated during the time period: Seat and Basin. Seat is activated for a longer time period. This makes the model predict "Spare_Time/TV" although the person is grooming in the bathroom.	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
I am not sure but I guess because of the previous seat firings. The model seems to be rather undecided between Grooming, / Spartime/TV / Other	Uncertain/unclear	No clear answer	Not clear / unsure	No clear answer	Participant unsure/does not understand; When	Data	
since the seat sensors are more active	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
High contribution of spiking seat sensor (and bias) in the first half (middle) to prediction of >spare_time/tv<. Even though there is a high contribution of the spiking basin (bias) sensor to label >grooming< towards the end, the confidence for >spare_time/TV< is still higher.	Both features and probability	Data and confidence	confidence	Classification confidenc	Seat used for long time (longer than basin)	Data	
[This was my initial reasoning]This is not clear to me. At 4918, the basin seems to have higher importance in the feature attribution. So, I am not sure why would the predicted class be spare_time/TV. Does the model take the whole sequential information and then make a prediction? Then, it can be that there is a long seating time and then the use of basin. So, at the end the output at 4918 is spare_time/TV. [After some clarification from Elisa about the working of SNN] I think the model weighs the feature importance in the whole sequence for taking the final decision at the last time step. May be also weighing both the bias and the seat sensor for the decision.	Uncertain & feature attribution (researcher intervention)	Data	Many explanations: sensor activation, confidence, bias, time	Data and confidence	Seat used for long time (longer than basin)	Data	
I assume that the system was trained with some training data set and this infers the average time in the toilet. Therefore, after being in the toilet it assumes going back to the last activity. However, I'm a little bit confused as there is a time frame without any sensor activated before being in the toilet; this makes me wonder what the true labels of these ones should be.	Personal interpretation	Learned patterns	Learned patterns during training	Learned patterns	Use of toilet during spare time was still classified as spare time in training data	Learned patterns	
Most of the 60s the sensor "seat" was activated and the "Basin" just for some seconds. maybe not enoght to switch the class to "Grooming"	Only feature attribution	Data	Sensor activation, time, seat, basin	Data	Seat used for long time (longer than basin)	Data	
~~The maindoor is active, which likely means the subject left the house, apparently that means to go get lunch according to the network. It is also likely that the subject left the house because nothing else in the house is active.~~	~~User error~~		~~Main door~~		~~Active main door = lunch; Model biased to choose lunch~~		User error (read cupboard activation as maindoor) but this means that the grid is not
We can see on the lower part that the timestep is blue, while it is also blue on the upperpart.	Both features and probability	Data and confidence	colour(s) in the figures/plots	Data and confidence	Color (Because model says so)	Data	
The cupboard was opened	Only feature attribution	Data	Opening the cupboard / use of cupboard	Data	Cupboard sensor activation indicates lunch	Data	
Because the cupboard was opened	Only feature attribution	Data	Opening the cupboard / use of cupboard	Data	Cupboard sensor activation indicates lunch	Data	
The cupboard was briefly open with a strong spike at the end and no other features (except bias) seemed to have activity, which relates to someone taking lunch out of the cupboard.	Only feature attribution	Data	Opening the cupboard / use of cupboard	Data	Cupboard sensor activation indicates lunch	Data	
the sensor is activated at the cupboard	Only feature attribution	Data	Sensor activation	Data	Cupboard sensor	Data	
the person was using the cupboard, which seems to be used when having lunch. (Blue marking at the end of he time period.)	Only feature attribution	Data	Opening the cupboard / use of cupboard	Data	Cupboard sensor activation indicates lunch	Data	
Because the cupboard was opened	Only feature attribution	Data	Opening the cupboard / use of cupboard	Data	Cupboard sensor activation indicates lunch	Data	
The cupboard sensor was triggered.	Only feature attribution	Data	Sensor activation	Data	Cupboard sensor	Data	
At timestep 184970 the cupboard sensor was firing	Only feature attribution	Data	Sensor activation	Data	Cupboard sensor	Data	
Because the cupboard sensor was activated	Only feature attribution	Data	Sensor activation	Data	Cupboard sensor	Data	
similar to previous example. there is a direct connection between cupboard to lunch.	Only feature attribution	Data	Opening the cupboard / use of cupboard	Data	Cupboard sensor activation indicates lunch	Data	
feature attribution shows the sensors is of cupboard, i suppose it directs to the lunch-related	Only feature attribution	Data	Sensor activation	Data	Cupboard sensor activation indicates lunch	Data	
The cupboard is used at timestep 184970, and there was is no other sensor activated before, so the model correctly predicts that the person eats lunch.	Only feature attribution	Data	Sensor activation	Data	Cupboard sensor activation indicates lunch	Data	

Although there are different classes predicted for this timeframe, the confidence distribution at timestep 184970 of lunch class is the highest among other classes. Thus, the model chose the one with the highest distribution to be the result of prediction.	Only probability distribution	Classification confidence	confidence	Classification confidenc	Lunch has the highest confidence	Classification Confidence	
Again, the confidence of the class Lunch outperforms the probability of other classes	Only probability	Classification confidence	confidence	Classification confidenc	Lunch has the highest	Classification	
similar to above answer, in step 184970, we can observe a spike, in which, the Lunch class has the highest confidence among the others	Both features and probability	Data and confidence	sensor activation and confidence	Data and confidence	Lunch has the highest confidence	Classification Confidence	
The Cupboard Sensor was activated at that point in time (184970). This was given as an input to the model and the model retuned the highest probability for the class label "Lunch". Second probable class was breakfast in this case, but that does not affect the final prediction as only the label with highest probability is considered.	Both features and probability	Data and confidence	Sensor activation	Data	Lunch has the highest confidence; Cupboard sensor activation indicates lunch	Data and confidence	
Since only the cupboard and bias sensor were activated at timestep 184970, the confidence distribution shows the highest result for "Lunch". Therefore, the model predicted "Lunch".	Both features and probability	Data and confidence	sensor activation and confidence	Data and confidence	Lunch has the highest confidence; Cupboard sensor activation	Data and confidence	
Because it has the highest confidence mainly due to the firing of the cupboard sensor.	Both features and probability	Data and confidence	sensor activation and confidence	Data and confidence	Lunch has the highest confidence; Cupboard	Data and confidence	
The sensor cupboard seems to be activated, so the person might be getting something to eat. As seen in the confidence distribution the majority of the activated classes are in line with that. Why the lunch class is predicted is unclear, as it might as well be breakfast or snack without any more information.	Both features and probability	Data and confidence	sensor activation and confidence + unsureness about model behavior	Data and confidence	Lunch has the highest confidence; Model biased to choose lunch; Cupboard sensor	Learned patterns	
The lunch class has the highest posterior probability / highest classification confidence. By the maximum a posteriori classification rule, the classifier would choose the lunch class as the most plausible label. I must say, however, I am unsure (based on the feature attribution window), why the confidence distribution is spread out so broadly across other classes. Is it due to the bias term?	Both features and probability	Data and confidence	confidence	Classification confidenc	Lunch has the highest confidence; Unsure due to confidence distribution (Confidence distribution is spread out between	Data and confidence	
Till around 184962s, the bias sensor is the only sensor which is activated. Until this time, the model seems to randomly predict classes with low confidence. However, after this the cupboard sensor is activated along with the bias sensor. The feature attribution shows that the cupboard sensor contributes to the model predicting the activity as Lunch with fairly high confidence, and this is the predicted class at the end of the time series.	Both features and probability	Data and confidence	sensor activation and confidence	Data and confidence	Lunch has the highest confidence; Unsure due to confidence distribution (Confidence distribution is spread out between classes); Cupboard	Data and confidence	
Well, I think it all depends on the training data set. And I also assume that these sets don't come from some Spaniard or Frenchy... it should have required at least 30 minutes activation from the Cooktop sensor I guess...	Personal interpretation	Learned patterns	Model training	Learned patterns	Model biased to choose lunch	Learned patterns	
Almost nothing happens in the timeframe, only in the end the cupboard is used. The model relates the usage of the cupboard for the most part to the eating actions but it seems that the sole use of the cupboard was more often recorded during lunch than during the other eating actions in the training data.	Only feature attribution	Data	Use of cupboard and model training	Learned patterns	Model biased to choose lunch; Cupboard sensor activation indicates lunch	Learned patterns	
The sensor cupboard was fired for a bit of time. You need that in order to make lunch however it could also have been breakfast, snack or dinner in my opinion.	Only feature attribution	Data	sensor activation and confidence + unsureness	Data and confidence	Model biased to choose lunch; Cupboard sensor	Learned patterns	
it considered the time	Used time of day as the only feature?	Learned patterns	Time	Learned patterns	Model considers the time of day	Learned patterns	
Cupboard sensor is activated, it means someone preparing something to eat. Since the sensor is know the time, it can predict the correct meal.	Only feature attribution	Data	Time and sensor activation	Data	Model considers the time of day; Cupboard sensor	Learned patterns	
sensor "Cupboard" seems to have a big influence for the label "lunch", so mabye the person always uses the cupboard for lunch and repect to the timestamp (bc the person used to have lunch around that time)	Both features and probability	Data and confidence	Using cupboard and time	Data	Model considers the time of day; Cupboard sensor activation indicates lunch	Learned patterns	
Reason for class Lunch: The cupboard door getting activated can point towards lunch. But it can also point towards other eating classes like breakfast, snack, dinner. That's the reason why there is also some prediction probability of the eating classes. Some comments: But I am not sure why the fridge, microwave or cooktop was not activated if the class is lunch. May be this just varies between different instances in the dataset.	Both features and probability	Data and confidence	sensor activation and confidence + unsureness about model behavior	Data and confidence	Unsure as other kitchen sensors are being inactive (Kitchen appliances are not used); Model biased to choose lunch; Cupboard sensor activation indicates lunch	Learned patterns	
it seems that the model predicts "lunch" class from the cupboard detection at the end. It looks like there are a bit of other probability that comes up when there is no detection (?).	Both features and probability	Data and confidence	Opening the cupboard / use of cupboard	Data	Unsure due to confidence distribution (Confidence distribution is spread out between	Data	
spiking cupboard sensor towards the end, which seems to be associated with label >lunch<, since it contributes (seemingly) exclusively to it. Small contribution of bias sensor to other labels still.	Both features and probability	Data and confidence	Sensor activation	Data	Unsure due to confidence distribution (Confidence distribution is spread out between	Data	
The cupboard is being used, probably at a time that is usually used for having lunch. It probably also be because the other kitchen appliences (fridge, microwave, cooktop, toaster) are not being used. It is unclear which of these reasons (or both) is the reason for lunch having the highest confidence score.	Both features and probability	Data and confidence	Using cupboard and time	Data	kitchen sensors being inactive (Kitchen appliances are not used); Model considers the time	Learned patterns	
Same as question 1. Because of the activation of the seat ot timestep 60	Only feature attribution	Data	Using seat	Data	Seat is used	Data	
Because only the seat was active the entire time	Only feature attribution	Data	Using seat	Data	No other sensors	Data	

Because there is only 1 feature called 'seat' recorded. Second, there is a strong spike at timestep 87083 that makes the confidence distribution 100% for spare_time/TV. This could mean that at the second 87083, it is sure that the sensor is activated because the person is siting.on / going very close by to that chair	Both features and probability	Data and confidence	Sensor activation and confidence	Data and confidence	High confidence; No other sensors activated; Seat is used	Data and confidence	
As far as I can see, virtually all feature attributions observed in the time window are associated with "spare time". Hence, it is not very surprising to see that the confidence distribution is almost fully concentrated at the spare time class. Given that this class has the highest posterior probability, the classifier predicts "spare time".	Both features and probability	Data and confidence	Confidence	Classification confidenc	High confidence; Seat is used	Data and confidence	
The only sensor that is being triggered is the one on the Seat, which means that the person is probably sitting there and therefore the model very confidently (and correctly) predicts 'Spare Time/TV' as the activity.	Both features and probability	Data and confidence	Sensor activation and confidence	Data and confidence	High confidence; No other sensors activated; Seat is used	Data and confidence	
Seat sensor have been activated for a while, so it predicts someone have spare time or watching TV.	Only feature attribution	Data	Sensor activation and time	Data	Seat is used	Data	
The seat sensor was activated, so I assume that is the seat in front of the TV, which makes sense for the Spare_Time/TV	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
Major confidence, almost 100% certainty on the class Spare_Time/tv	Only probability	Classification confidence	Confidence	Classification confidenc	High confidence	Classification	
Because the sensor at the seat is activated for certain duration.	Only feature attribution	Data	Sensor activation and	Data	Seat is used	Data	
spare time is predicted as the person is sensed at the seat. This means that the seat relates to the spare time class	Only feature attribution	Data	Using seat	Data	Seat is used	Data	
There was only seating recorded in the given timeframe which apparently directly relates to the spare_time/TV action. Other actions could record seating as well but probably another action will be recorded as well. For example dinner would probably include the fridge.	Only feature attribution	Data	Using seat	Data	No other sensors activated; Seat is used	Data	
The Seat Sensor was activated which resulted in very high (almost 100% confidence) probability for "Spare_Time/TV"	Both features and probability	Data and confidence	Sensor activation and confidence	Data and confidence	High confidence; Seat is used	Data and confidence	
The seat sensor is activated and this has a very high confidence, so this is likely leading to someone watching TV.	Both features and probability	Data and confidence	Sensor activation and confidence	Data and confidence	High confidence; Seat is used	Data and confidence	
In this example, the bias and seat sensors are shown as spiking at each time step. Since only the seat sensor is activated continuously, the model starts predicting the activity as Spare_Time/TV around 87033s. The sensor activity remains unchanged for the entire duration shown, and probably due to this, the model shows high confidence at the last time step.	Both features and probability	Data and confidence	Sensor activation, confidence and time	Data and confidence	High confidence; No other sensors activated; Seat is used	Data and confidence	
We can see on the bottom it is pink as well as the upper part.	Both features and	Data and confidence	Color(s) in the figures	Data and confidence	Because model says so	Data	
The subject is in the seat for an extended period of time without activating anything else, the most likely corresponding class is "spare time/TV", because one can do that from the seat without interaction with the other sensors	Only feature attribution	Data	Sensor activation	Data	No other sensors activated; Seat is used	Data	
Because the person was sitting the whole time, with the last time period contibuting most to this decision.	Only feature attribution	Data	Using seat	Data	Seat is used	Data	
Because the seat was used	Only feature attribution	Data	Using seat	Data	Seat is used	Data	
Because the seat sensor was constantly activated	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
The model predicted "Spare_Time/TV" mainly because the seat sensor was activated at timestep 87083.	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
The sensor seat was fired. And the logically activity would be spare time/ TV when sitting on the seat.	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
The seat sensor was firing for the timestep	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
Because the seat sensor was activated	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
it looks like a direct seat sensor to spare_time connection.	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
feature attributions shows it involves the sensor of seat, maybe another feature with enough time duration, so it predicts to the spare_time	Only feature attribution	Data	Sensor activation and time	Data	Seat is used	Data	
in step 87083, we can observe a spike, in which, the Spare_Time class has the highest confidence among the others	Only probability distribution	Classification confidence	Confidence	Classification confidenc	High confidence	Classification Confidence	
At timestep 87083, the seat sensor is activated and the model correctly predicts that the person is watching TV in their spare time.	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
Because of the firing of the Seat sensor.	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
only seat sensor is active and receiving solid data	Only feature attribution	Data	Sensor activation	Data	No other sensors	Data	
high contribution of spiking seat sensor, in particular at later timesteps. likely, the seat sensor is associated with >spare_time/TV<. Further, bias sensor seems to contribute to >spare_time/TV< as well in this case.	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
There is a long duration of seating which implies that the person is watching TV and the model gives importance to the seat feature at 87083 time instance.	Only feature attribution	Data	Using seat and time	Data	Seat is used	Data	
There is a constant activated input from sensor Seat, so the most logical prediction is to continue without any change	Only feature attribution	Data	Sensor activation	Data	Seat is used	Data	
Bc sensor "seat" was activated for last 60s, no other sensor (except bais) --> person is seated for longer time and is not doing anything else	Only feature attribution	Data	Sensor activation and time	Data	No other sensors activated; Seat is used	Data	

Fleiss' Kappa calculation

	Clusters					
Survey responses (Background colour indicates Question)	**Data**	**Classification Confidence**	**Data and Confidence**	**Learned patterns**	**No certain answer**	**P_i**
Because the seat was activated longer than the bassin and the confidence is nearly twice as high for tv than for grooming	2	0	1	0	0	0,333333333
Because the first half of the time was spent on the seat	3	0	0	0	0	1
The big distribution of class spare_time is influenced by the long spanning and more bias of the feature seat, which makes this class distribution have the highest value and thus the model chose this class. On the other hand, another active sensor was spiked at basin sensor (which indicates a stronger detection) contributes to class grooming to be activated. However, the confidence distribution of class grooming is not as high as the spare_time class so the model could not choose it. This can indicate that seat feature is seen as some sort of noises and together with a big amount of noises, the result will be influenced	1	0	2	0	0	0,333333333
Similar to the previous answers, I would once again base this on the highest classification confidence.	0	3	0	0	0	1
Because the person was sitting on the seat first, which lasted longer in this timeframe than going the the basin.	3	0	0	0	0	1
He/she spent most of the time on seat according to sensor, but then it appears that he/she decided to leave and basin sensor is activated at timestep 4918. So it should be grooming.	3	0	0	0	0	1
The seat was active with the most spikes and therefore the model predicts that. Still, the confidence is only 0.4 of the model. However, the true label was only the third largest indicator, probably because it was only briefly active. Not sure about the 'Other' category though.	1	0	2	0	0	0,333333333
the confidence is higher for Spare_Time then Grooming for timestep 4918	0	3	0	0	0	1
perhaps, the duration at the seat is longer than at the toilet. So it predicted with higher confidence to Spare time class	2	0	1	0	0	0,333333333
detection at seat chich relates to spare time appears more, and in this case has a higher probability compared to the basin, which corresponds to grooming.	2	0	1	0	0	0,333333333
Until second 4877, the seat sensor is spiking. After that, no sensors record any spiking until second 4912. From there until the end, the basin sensor records an action. The model predicts four classes, with the class "spare_time/TV" having the highest prediction probability of a bit more than 0.4. The true label is grooming. The label was most likely predicted because the seating action is active longer than the basin action and seating probably relates to the spare_time/TV class.	2	0	1	0	0	0,333333333

First the seat sensor was activated and shortly after that, up until the time of interest (4918) the basin was activated. The model learned during training that this is most probably related to the activity "Grooming". However in this case the prediction is wron as it differs from the true label.	1	0	0	1	1	0
In this window the largest part of the time the seat is activated, so overall the most plausible action would be watching TV. At the last few time steps the activation is indicating that the basin is activated which lead to grooming, which would be the most logical at that current time step. As seen the confidence of grooming is lower than the spare time/TV, probably due to the fact that the latter has a longer duration.	2	0	1	0	0	0,333333333
For the first part of the time series, the seat sensor is activated along with the bias sensor. This leads the model to predict the activity as Spare_Time/TV. However, towards the end the basic sensor is activated, while the seat sensor is not. However for the given time series, the feature attribution graph shows that the higher confidence for Spare_Time/TV could be attributed to the longer duration that the seat sensor was activated, which usually corresponds to Spare_Time/TV as seen from previous examples. It is unclear why the confidence of Other is so high when the feature attribution graph does not clearly show this.	1	0	2	0	0	0,333333333
I think that the bottom might be the the sensor data throughout the time now. So that we can see that grooming is happening at 4918. But the upperpart shows the predictions and shows that spare_time/tv is predicted since it has the highest p(C|x)	1	1	1	0	0	0
It is not clear to me wether the prediction is about the final time step or the whole time interval. If it is about the whole time interval it predicts spare_time/TV because that is predictablr for most of the time. If it is only for the final time step, the SNN is slow in response to a new input and therefore still puts spare_time/tv on top while for the later period the basin sensor was active.	1	0	0	0	2	0,333333333
Because the majority of the time, the person was sitting which seems to indicate spare time TV.	3	0	0	0	0	1
Because the seat was used	3	0	0	0	0	1
More seat activation then Basin.	3	0	0	0	0	1
At timestep 4918 only bias and basin sensors were activated which would normally result in the prediction "Grooming". Nevertheless, the model predicted "Spare_Time/TV" because the seat sensor was activated longer than the basin sensor but shortly before its activation. As a result the probability for the label "Spare_Time/TV" to be true is still higher than for "Grooming".	2	0	1	0	0	0,333333333
Most time was spend on the seat.	3	0	0	0	0	1
Because the seat sensor was firing/spiking. Looks like they got up to pee so P wasn't 1?	3	0	0	0	0	1
Not clear to me		0	0	0	3	1
same as before, it looks like there is a direct connection between the seat sensor and sparetime. Model bias I guess.	2	0	0	1	0	0,333333333

the duration for activated sensor of seat is longer than the duration of basin in the past?	3	0	0	0	0	1
similar to above answer, in step 4918, we can observe a spike, in which, the Spare_Time class has the highest confidence among the others		3	0	0	0	1
In this case, there are two different sensors activated during the time period: Seat and Basin. Seat is activated for a longer time period. This makes the model predict "Spare_Time/TV" although the person is grooming in the bathroom.	3	0	0	0	0	1
I am not sure but I guess because of the previous seat firings. The model seems to be rather undecided between Grooming, / Spartime/TV / Other	1	0	0	0	2	0,333333333
since the seat sensors are more active	3	0	0	0	0	1
High contribution of spiking seat sensor (and bias) in the first half (middle) to prediction of >spare_time/tv<. Even though there is a high contribution of the spiking basin (bias) sensor to label >grooming< towards the end, the confidence for >spare_time/TV< is still higher.	1	1	1	0	0	0
[This was my initial reasoning]This is not clear to me. At 4918, the basin seems to have higher importance in the feature attribution. So, I am not sure why would the predicted class be spare_time/TV. Does the model take the whole sequential information and then make a prediction? Then, it can be that there is a long seating time and then the use of basin. So, at the end the output at 4918 is spare_time/TV. [After some clarification from Elisa about the working of SNN] I think the model weighs the feature importance in the whole sequence for taking the final decision at the last time step. May be also weighing both the bias and the seat sensor for the decision.	2	0	1	0	0	0,333333333
I assume that the system was trained with some training data set and this infers the average time in the toilet. Therefore, after being in the toilet it assumes going back to the last activity. However, I'm a little bit confused as there is a time frame without any sensor activated before being in the toilet; this makes me wonder what the true labels of these ones should be.	0	0	0	3	0	1
Most of the 60s the sensor "seat" was activated and the "Basin" just for some seconds. maybe not enoght to switch the class to "Grooming"	3	0		0	0	1
We can see on the lower part that the timestep is blue, while it is also blue on the upperpart.	1	0	2	0	0	0,333333333
The cupboard was opened	3	0	0	0	0	1
Because the cupboard was opened	3	0	0	0	0	1
The cupboard was briefly open with a strong spike at the end and no other features (except bias) seemed to have activity, which relates to someone taking lunch out of the cupboard.	3	0	0	0	0	1
the sensor is activated at the cupboard	3	0	0	0	0	1
the person was using the cupboard, which seems to be used when having lunch. (Blue marking at the end of he time period.)	3	0	0	0	0	1

Because the cupboard was opened	3	0	0	0	0	1
The cupboard sensor was triggered.	3	0	0	0	0	1
At timestep 184970 the cupboard sensor was firing	3	0	0	0	0	1
Because the cupboard sensor was activated	3	0	0	0	0	1
similar to previous example. there is a direct connection between cupboard to lunch.	3	0	0	0	0	1
feature attribution shows the sensors is of cupboard, i suppose it directs to the lunch-related	3	0	0	0	0	1
The cupboard is used at timestep 184970, and there was is no other sensor activated before, so the model correctly predicts that the person eats lunch.	3	0	0	0	0	1
Although there are different classes predicted for this timeframe, the confidence distribution at timestep 184970 of lunch class is the highest among other classes. Thus, the model chose the one with the highest distribution to be the result of prediction.	0	3	0	0	0	1
Again, the confidence of the class Lunch outperforms the probability of other classes	0	3	0	0	0	1
similar to above answer, in step 184970, we can observe a spike, in which, the Lunch class has the highest confidence among the others	0	1	2	0	0	0,333333333
The Cupboard Sensor was activated at that point in time (184970). This was given as an input to the model and the model retuned the highest probability for the class label "Lunch". Second probable class was breakfast in this case, but that does not affect the final prediction as only the label with highest probability is considered.	1	0	2	0	0	0,333333333
Since only the cupboard and bias sensor were activated at timestep 184970, the confidence distribution shows the highest result for "Lunch". Therefore, the model predicted "Lunch".	0	0	3	0	0	1
Because it has the highest confidence mainly due to the firing of the cupboard sensor.	0	0	3	0	0	1
The sensor cupboard seems to be activated, so the person might be getting something to eat. As seen in the confidence distribution the majority of the activated classes are in line with that. Why the lunch class is predicted is unclear, as it might as well be breakfast or snack without any more information.	0	0	2	1	0	0,333333333
The lunch class has the highest posterior probability / highest classification confidence. By the maximum a posteriori classification rule, the classifier would choose the lunch class as the most plausible label. I must say, however, I am unsure (based on the feature attribution window), why the confidence distribution is spread out so broadly across other classes. Is it due to the bias term?	0	1	2	0	0	0,333333333
Till around 184962s, the bias sensor is the only sensor which is activated. Until this time, the model seems to randomly predict classes with low confidence. However, after this the cupboard sensor is activated along with the bias sensor. The feature attribution shows that the cupboard sensor contributes to the model predicting the activity as Lunch with fairly high confidence, and this is the predicted class at the end of the time series.	0	0	3	0	0	1

Well, I think it all depends on the training data set. And I also assume that these sets don't come from some Spaniard or Frenchy... it should have required at least 30 minutes activation from the Cooktop sensor I guess...	0	0	0	3	0	1
Almost nothing happens in the timeframe, only in the end the cupboard is used. The model relates the usage of the cupboard for the most part to the eating actions but it seems that the sole use of the cupboard was more often recorded during lunch than during the other eating actions in the training data.	1	0	0	2	0	0,333333333
The sensor cupboard was fired for a bit of time. You need that in order to make lunch however it could also have been breakfast, snack or dinner in my opinion.	1	0	1	1	0	0
it considered the time	0	0	0	3	0	1
Cupboard sensor is activated, it means someone preparing something to eat. Since the sensor is know the time, it can predict the correct meal.	2	0	0	1	0	0,333333333
sensor "Cupboard" seems to have a big influence for the label "lunch", so mabye the person always uses the cupboard for lunch and repect to the timestamp (bc the person used to have lunch around that time)	1	0	1	1	0	0
Reason for class Lunch: The cupboard door getting activated can point towards lunch. But it can also point towards other eating classes like breakfast, snack, dinner. That's the reason why there is also some prediction probability of the eating classes. Some comments: But I am not sure why the fridge, microwave or cooktop was not activated if the class is lunch. May be this just varies between different instances in the dataset.	0	0	2	1	0	0,333333333
it seems that the model predicts "lunch" class from the cupboard detection at the end. It looks like there are a bit of other probability that comes up when there is no detection (?).	2	0	1	0	0	0,333333333
spiking cupboard sensor towards the end, which seems to be associated with label >lunch<, since it contributes (seemingly) exclusively to it. Small contribution of bias sensor to other labels still.	2	0	1	0	0	0,333333333
The cupboard is being used, probably at a time that is usually used for having lunch. It probably also be because the other kitchen appliences (fridge, microwave, cooktop, toaster) are not being used. It is unclear which of these reasons (or both) is the reason for lunch having the highest confidence score.	1	0	1	1	0	0
Same as question 1. Because of the activation of the seat ot timestep 60	3	0	0	0	0	1
Because only the seat was active the entire time	3	0	0	0	0	1
Because there is only 1 feature called 'seat' recorded. Second, there is a strong spike at timestep 87083 that makes the confidence distribution 100% for spare_time/TV. This could mean that at the second 87083, it is sure that the sensor is activated because the person is siting.on / going very close by to that chair	0	0	3	0	0	1

As far as I can see, virtually all feature attributions observed in the time window are associated with "spare time". Hence, it is not very surprising to see that the confidence distribution is almost fully concentrated at the spare time class. Given that this class has the highest posterior probability, the classifier predicts "spare time".	0	1	2	0	0	0,333333333
The only sensor that is being triggered is the one on the Seat, which means that the person is probably sitting there and therefore the model very confidently (and correctly) predicts 'Spare Time/TV' as the activity.	0	0	3	0	0	1
Seat sensor have been activated for a while, so it predicts someone have spare time or watching TV.	3	0	0	0	0	1
The seat sensor was activated, so I assume that is the seat in front of the TV, which makes sense for the Spare_Time/TV	3	0	0	0	0	1
Major confidence, almost 100% certainty on the class Spare_Time/tv		3	0	0	0	1
Because the sensor at the seat is activated for certain duration.	3	0	0	0	0	1
spare time is predicted as the person is sensed at the seat. This means that the seat relates to the spare time class	3	0	0	0	0	1
There was only seating recorded in the given timeframe which apparently directly relates to the spare_time/TV action. Other actions could record seating as well but probably another action will be recorded as well. For example dinner would probably include the fridge.	3	0	0	0	0	1
The Seat Sensor was activated which resulted in very high (almost 100% confidence) probability for "Spare_Time/TV"	0	0	3	0	0	1
The seat sensor is activated and this has a very high confidence, so this is likely leading to someone watching TV.	0	0	3	0	0	1
In this example, the bias and seat sensors are shown as spiking at each time step. Since only the seat sensor is activated continuously, the model starts predicting the activity as Spare_Time/TV around 87033s. The sensor activity remains unchanged for the entire duration shown, and probably due to this, the model shows high confidence at the last time step.	0	0	3	0	0	1
We can see on the bottom it is pink as well as the upper part.	1	0	2	0	0	0,333333333
The subject is in the seat for an extended period of time without activating anything else, the most likely corresponding class is "spare time/TV", because one can do that from the seat without interaction with the other sensors	3	0	0	0	0	1
Because the person was sitting the whole time, with the last time period contibuting most to this decision.	3	0	0	0	0	1
Because the seat was used	3	0	0	0	0	1
Because the seat sensor was constantly activated	3	0	0	0	0	1
The model predicted "Spare_Time/TV" mainly because the seat sensor was activated at timestep 87083.	3	0	0	0	0	1
The sensor seat was fired. And the logically activity would be spare time/ TV when sitting on the seat.	3	0	0	0	0	1
The seat sensor was firing for the timestep	3	0	0	0	0	1
Because the seat sensor was activated	3	0	0	0	0	1
it looks like a direct seat sensor to spare_time connection.	3	0	0	0	0	1

feature attributions shows it involves the sensor of seat, maybe another feature with enough time duration, so it predicts to the spare_time	3	0	0	0	0	1
in step 87083, we can observe a spike, in which, the Spare_Time class has the highest confidence among the others	0	3	0	0	0	1
At timestep 87083, the seat sensor is activated and the model correctly predicts that the person is watching TV in their spare time.	3	0	0	0	0	1
Because of the firing of the Seat sensor.	3	0	0	0	0	1
only seat sensor is active and receiving solid data	3	0	0	0	0	1
high contribution of spiking seat sensor, in particular at later timesteps. likely, the seat sensor is associated with >spare_time/TV<. Further, bias sensor seems to contribute to >spare_time/TV< as well in this case.	3	0	0	0	0	1
There is a long duration of seating which implies that the person is watching TV and the model gives importance to the seat feature at 87083 time instance.	3	0	0	0	0	1
There is a constant activated input from sensor Seat, so the most logical prediction is to continue without any change	3	0	0	0	0	1
Bc sensor "seat" was activated for last 60s, no other sensor (except bais) --> person is seated for longer time and is not doing anything else	3	0	0	0	0	1
Total	181	26	60	19	8	
P_j	0,615646259	0,088435374	0,204081633	0,06462585	0,027210884	

p_bar	0,768707483
p_bar_e	0,433407377
kappa	0,591783395