ARIEL UNIVERSITY

MASTER THESIS

Meta Learning Based Deception Detection From Speech

Author:
Noa Mansbach

Supervisor:
Prof. Amos Azaria

Department of Computer Science

December 3, 2022

ARIEL UNIVERSITY

MASTER THESIS

Meta Learning Based Deception Detection From Speech

Author:
Noa Mansbach

Supervisor:
Prof. Amos Azaria

Department of Computer Science

December 3, 2022

Declaration of Authorship

I, Noa Mansbach, hereby declare that this thesis proposal entitled, "Meta Learning Based Deception Detection From Speech" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Dedicated to the memory of my beloved father, who always encouraged me to learn and strive for excellence…

Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor Prof. Amos Azaria for giving me the opportunity to do research and providing guidance throughout this research with a lot of patience.
I would like to thank my husband and my family for their support and help with everything.
I would like to thank Ariel University for the fellowship which support me while conducting my research.

This work was supported in part by the Ministry of Science and Technology of Israel.

Ariel University

Abstract

Faculty of Natural Sciences
Department of Computer Science

Master

Meta Learning Based Deception Detection From Speech

by Noa Mansbach

It is difficult to overestimate the importance of detecting human deception, specifically by using speech cues. While several works attempt to detect deception from speech, most do not separate training samples from test samples by the people who said each statement or by the environments in which each sample was recorded. This may result in less reliable detection results. In this paper, we take a meta-learning approach in which a model is trained on a variety of learning tasks to enable it to solve new learning tasks using only a few samples. In our approach, we split the data according to the persons (and recording environment), i.e., some people are used for training, and others are used for testing only, but we do assume a few labeled samples for each person in the data set. We introduce CHAML, a novel deep learning architecture that receives as input the sample in question along with two more truthful samples and non-truthful samples from the same person. We show that our method outperforms other state-of-the-art methods of deception detection based on speech and other approaches for meta-learning.

Contents

Declaration of Authorship **i**

Acknowledgements **iii**

Abstract **iv**

1 Introduction **1**

2 Related Work **3**
2.1 Deception Detection 3
2.2 Speech Emotion Recognition 4
2.3 Few-Shot Learning 5

3 Data Collection **7**

4 Comparative Hint Approach Meta-Learning (CHAML) **9**
4.1 Embedding Types 9
4.1.1 Five Sound Features 9
4.1.2 Wav2Vec 2.0 9
4.2 CHAML - Core 10
4.3 FSFM Classifier 11

5 Evaluation **14**
5.1 Baseline Methods 14
5.1.1 FSFM - Fine-Tuning 14
5.1.2 Prototypical Network 14
5.1.3 Model-Agnostic Meta-Learning (MAML) 14
5.2 Results 15

6 Conclusions & Future work **17**

List of Publications

Published

1. Mansbach Noa, Neiterman Hershkovitch Evgeny, and Azaria Amos.
   An Agent for Competing with Humans in a Deceptive Game Based on Vocal Cues, Interspeech 2021.

Submitted

2. Mansbach Noa and Azaria Amos.
   Meta Learning Based Deception Detection From Speech, Applied Sciences, 2022.

3. Alouch Merav, Mansbach Noa, Azaria Amos, and Azoulay Rina.
   Utilizing Machine Learning for Detecting Harmful Situations by Audio and Text, IEEE Access, 2022.

List of Figures

3.1 The “Cheat-Game" interface. 7

4.1 An illustration of CHAML Model. 11
4.2 An Illustration of FSFM Model. 12

List of Tables

3.1 Subjects' demographic information 8
3.2 Data-set Division 8

4.1 Hyper-parameters used for fine-tuning 10
4.2 Support-Query Division 11

5.1 Comparison of models performance 15
5.2 CHAML - Wav2Vec 2.0 performance on the query set of the test samples 15
5.3 CHAML vs. meta-learning methods Average Computation Times . . . 16

List of Algorithms

1 CHAML Training Process 13

List of Abbreviations

CHAML	**C**omparative **H**int **A**pproach **M**eta **L**earning
SER	**S**peech **E**motion **R**ecognition
MAML	**M**odel **A**gnostic **M**eta **L**earning
ASR	**A**utomatic **S**peech **R**ecognition
FSFM	**F**ive **S**ound **F**eatures **M**odel

Chapter 1

Introduction

Lying and deception are an inherent part of human nature. While some lies are considered small and may even be helpful for a smoother interaction between humans, others may be devastating and cause major damage. However, despite deception detection being essential to everyone in their daily life, it is challenging for humans to determine whether a person is being deceptive [EF03, DSHW10]. Therefore, throughout history, many methods and devices were developed for that task [Tro38], and more recently, machine learning methods based on text, video, and speech [OCCH11, HS21, PRAMB15].

Typically, works for speech deception detection do not separate train and test based on the person so that an individual may have examples in both train and test [HS21]. Nor do most works split the training and test data according to the recording environment. This results in less reliable models, as in practice, the model must learn about some population in some recording environment and then be used to produce predictions for a different population in a different recording environment. Consequently, in this work, we split the data according to the persons, i.e., some people are used for training, and others are used for testing only.

One of the main difficulties in deception detection based on speech is to learn the features of lying from some people and in some recording environments and apply this knowledge to others, despite the fact that different people tend to lie differently. Therefore, in order to achieve high performance, the model is required to learn which speech-related features are specific to a person and which are general.

In addition, we consider a meta-learning approach for deception detection based on speech. For that, we assume that we have very few labeled samples for each person (namely, two positive samples and two negative ones). This approach is inspired by the well-known polygraph test [C+03], in which several comparison questions are asked at the beginning of the polygraph interview in order to obtain physiological measures of the subject when telling the truth and when lying. Namely, in our approach, the model is trained on a set of training tasks; each task represents a person from the subjects in our data set and consists of a support set used for learning about the task and a query set used to evaluate the performance of this task. The support set contains four examples from the person's samples, two positive samples and two negative ones, and the query set contains the remaining samples of the person.

Our approach to meta-learning differs from the typical one. In a typical meta-learning problem, there are typically several classes in each task, which differ from task to task. In addition, training and test tasks typically have different classes. However, in our setting, we have the same two classes for all tasks, but each task represents a different person from our data, and therefore, the data features are very

different between different tasks. We present an innovative method of deception detection in the meta-learning setting and show that it outperforms the existing state-of-the-art methods of deception detection based on speech and other approaches for meta-learning. In our method, we use the comparative hint approach, which gives the model hints about the new environment (in our case, a new person) by providing some true and false examples from the same person sample set together with the tested sample. Namely, our model processes the positive and negative pairs and combines the result of this process with the tested sample into a vector fed into a neural classifier.

We believe that by using our unique architecture, the model can compare the tested sample to the given hints, learn the person's way of lying, and improve its detection performance.

To summarize, our main contribution in this work is tackling the problem of deception detection based on audio signals when having different train and test environments (i.e., persons), and when the model is provided with very few true and false labeled samples for each person in the test-set. To that end, we gathered a massive amount of data and developed CHAML, a novel solution based on the meta-learning approach that uses samples for each person to learn their way of lying. This method outperforms state-of-the-art methods and can be applied to other environments that include different tasks, using any neural network as its classifier.

Chapter 2

Related Work

2.1 Deception Detection

The deception detection task has been explored in many types of research using different approaches and techniques: some based on text, some on speech, some on video, and others on physiological measures. In the beginning, most research was focused on analyzing physiological measures, such as breathing rate, heart rate, blood pressure, and body temperature [PR77, TO68, Vri00]. Other studies found the connection between deception and human behaviors [KHD06, DJ82, DLM+03].

However, detecting the physiological measures of a human requires special instruments and may be invasive and expensive, while humans may have difficulties identifying deception behaviors. Therefore, the use of machine learning methods is almost necessary. Many studies researched the use of machine learning methods for deception detection.

For detection based on text problems, Ott et al. [OCCH11] develop a corpus of deceptive reviews and use Näive Bayes and Support Vector Machine (SVM) as the classifiers utilizing Linguistic Inquiry and Word Count (LIWC) combined with bigrams. Feng et al. [FBC12] show that using Context Free Grammar (CFG) parse trees consistently improves detection performance. Barsever et al. [BSN20] use the BERT (Bidirectional Encoder Representations from Transformers) network and show that compared with truthful text, deceptive text tends to be more formulaic and less varied.

There have been only a few works that attempt to detect deception based on speech cues. Hirschberg et al. [HBB+05] develop a corpus of deceptive speech using one on one interviews. Nasri et al. [NOA16] use SVM model utilizing Mel Frequency Cepstral Coefficient (MFCC). Graciarena et al. [GSS+06] train a classifier using combined both linguistic features and acoustic features as a combination of MFCC and prosodic features and Marcolla et al. [MdSD20] use an LSTM neural network on a set of MFCC characteristics extracted from audio speech to deception detecting based on voice stress. Xie et al. [XLT+18] extract variable length frame-level speech features from different length's speech samples and use a recurrent neural network combined with a convolution operation as their model.

Other studies focus on deception detection in videos using different feature extraction methods such as IDT (Improved Dense Trajectory) feature, and high-level features represent facial micro-expressions extracted from the videos, with machine learning techniques [DZL+19, SHD+20, MM20, MMS+22]. Ding et al. [DZL+19] develop an automated deception detection model that consists of three main modules: face focused cross-stream network which deep joint feature learning from facial expressions and body motions for video, a meta-learning module, and an adversarial learning module that generates a 256-dimension feature vector for each synthesized video. The meta-learning module was used to deal with their data scarcity problem

by using pairwise comparison. Each deceptive video sample was combined with four true samples to generate five pairs. The model outputs the probability for each pair to be from the same class.

Other researchers apply a multi-modal approach for deception detection from video data sets. Pérez-Rosas et al. [PRAMB15] introduce a collected data-set consisting of videos collected from public court trials. They apply a multi-modal approach for deception detection on their data set using inputs from different modalities, i.e., video, audio, and text. Wu et al. [WSDS18] use common machine learning techniques such as SVM, Näive Bayes, Decision Trees, Random Forests, Logistic Regression, and Adaboost. They test different combinations of multi-modal features: IDT feature and high-level features represent facial micro-expressions extracted from the videos as their motion features, MFCC as the audio features and encoded the video transcripts using Glove (Global Vectors for Word Representation). Other researches [GAH17, KMPC18] introduce a deep learning multi-modal approach on the same data-set, using Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) models.

2.2 Speech Emotion Recognition

The speech classification task has many fields such as emotion recognition, speaker identification, language identification, etc. The Speech Emotion Recognition (SER) task is recognizing the emotional aspects of speech and classifying them into emotion categories. This task may be considered very close to our task, as an emotional aspect may be involved in people's lying.

At the beginning, most works on the SER problem proposed solutions based on classical ML methods such as Hidden Markov Models (HMM), SVM, and Random Forests. Nwe et al. [NFS03] propose a method that represents the speech signals and a discrete HMM as the classifier by using short time Log Frequency Power Coefficients (LFPC). The performance of the LFPC feature parameters is compared with that of the linear prediction Cepstral coefficients (LPCC) and MFCC feature parameters commonly used in speech recognition systems. Jain et al. [JNB+] use a SVM to classify the speech taken as one of the four emotions (sadness, anger, fear, and happiness). They classify these emotional states using two strategies: One against All (OAA) and Gender Dependent Classification. Noroozi et al. [FTDG17] propose a method using Random Forests for vocal emotion recognition. This technique uses Random Forests to represent the speech signals and the Decision-Trees approach to classify them into different categories.

Over the past several years, deep learning has become one of the main approaches for solving the SER problem [KJB+19]. Trigeorgis et al. [TRB+16] develop an end-to-end speech emotion recognition using a combination of CNN with LSTM networks for learning a representation of the speech signal directly from the raw time representation. Han et al. [HYT14] propose using deep neural networks to extract high-level features from raw data to solve the speech emotion recognition problem. They produce an emotion state probability distribution for each speech segment using deep neural networks. Fayek et al. [FLC17] propose a deep learning framework using CNNs and a spectrogram of a speech signal as the input. Another promised approach in various speech-related tasks is the Wav2Vec 2.0 framework [AHAM20]. This framework is used for self-supervised learning of vector representation from speech audio. Recent research has shown that the Wav2Vec 2.0 framework is also a robust alternative for SER and speaker identification tasks [PRF21, CR21, VVL22].

Therefore, we also use this framework for one of the extraction feature types in our task.

Recently, the few-shot learning method was applied to the SER problem. Guibon et al. [GLF+21] use Prototypical networks for emotion sequence labeling, Feng and Chaspari [FC21] use Siamese Neural Network for emotion recognition of spontaneous speech and [NS22] use MAML for solving the Multilingual SER problem. Hence, as this approach seems promising for the SER task, we adapt it to be used in our problem.

2.3 Few-Shot Learning

The method in our paper is based on the idea of the meta-learning framework or "learning to learn" [TP98], specifically in the field of few-shot learning. The meta-learning framework is based on learning from prior experience with other tasks, i.e., learning how to learn to classify given a set of training tasks such that the model can solve new learning tasks. One of the main challenges in the meta-learning framework is to train an accurate model using only a few training examples given prior experience with similar tasks. This is called few-shot learning.

Few-shot learning, is training a model for learning from very few samples and generalizing to many other new examples; most approaches for few-shot learning follow the meta-learning framework. It measures a model's ability to quickly adapt to new environments and tasks using only a few examples and training iterations. For that, the model is trained on a set of tasks in a meta-learning phase, allowing it to adapt quickly to new tasks with just a few examples. Each task consists of a support set used for learning how to solve each specific task and a query set containing further examples of each specific task, which are used to evaluate the performance on each task. A task can be utterly non-overlapping with another; the classes from one task may never appear in another. The model's performance is evaluated by the average test accuracy across the query sets of many testing tasks. As the meta-learning process proceeds, the model parameters are updated based on the training tasks. The loss function is derived from the classification performance on the query set of each of the training tasks based on the knowledge gathered from its support set. The network is given a different task at each time step, so it must learn to discriminate data classes in general rather than specific subsets.

A common way of attempting to solve a few-shot learning problem is by using prior knowledge about similarity. This is done by learning class embeddings that tend to separate classes even if they have never been seen before. One of the earlier methods for solving few-shot problems is a pairwise comparator [KZS+15, HA15] that was developed to classify two examples as belonging or not to the same class based on their similarity, even though the model had never seen those classes before. This method can be adapted to few-shot learning by classifying an example from the query set according to its maximum similarity to an example in the support set. A more elegant way is multi-class comparators [VBL+16, SSZ17], which learn a common representation for each class in the training set and match each new test example using cosine similarity. Snell et al. [SSZ17] propose Prototypical Networks, which average the embeddings of the examples from the same class to compute the class prototype (mean vector). Then a distance metric is used to calculate the similarity (a negative multiple of the Euclidean distance) between each query embeddings to each of the classes' prototypes for finding the most similar class.

Alternatively, the few-shot learning problem can be solved by learning parameters that generalize better to similar tasks and can be fine-tuned very quickly when applied to different tasks. An implementation of that approach is Model-Agnostic Meta-Learning (MAML), introduced by Finn et al. [FAL17]. The model is initialized with random weights, and iteratively, for each task in a meta-batch of tasks, it fine-tunes a copy learner using the weights of the primary model (meta-learner). The learner weights are updated using the loss from the query samples in the task by stochastic gradient descent. At the end of each training task, the losses and gradients from the queries are accumulated, the derivative of the mean loss concerning the primary model's weights is computed, and the weights in the primary model are updated. The primary model's weights improve during this process so that the model can fine-tune to other tasks faster.

Chapter 3

Data Collection

FIGURE 3.1: The “Cheat-Game" interface.

The data presented in this work is based on the “Cheat-Game”, which is a turn-taking card game. Each player is dealt 8 cards and the goal is to play all cards. The centered pile accumulates all cards played by the players, and every turn the value of the recently played card is supposed to either go up by one or down by one. On each turn, a player may place up to four cards (faced down) on the centered pile. The player then states which cards she disposed; however, the player may claim to put cards that are different from what she actually played (i.e., *a false claim*). Nevertheless, the player must claim to play cards that have a value of either one above or one below the recently played card(s). If a player suspects that her opponent is cheating, i.e., played cards that are different than what she has stated, the player may call out a cheat. If the opponent did in-fact cheat, she collects all the cards; otherwise, the player that called out a cheat collects the cards.

We use the implementation of Mansbach et al. [NHA21] for the “Cheat-Game” (see Figure 3.1 for a screenshot). During game-play, the claims of the players are recorded and added to our data set along with the actual cards played. This information allows us to determine whether a claim is true or false.

To obtain high-quality results, we improved the game environment by adding an audio test at the beginning of the game and an option for the player to hear her

claim so that she could make sure it sounds clear; if not, it could be re-recorded. Unfortunately, some players who played illegal cards (i.e., opted to cheat), rather than saying that they played legal cards, stated the cards they actually played. Such statements cannot be seen as an untruthful statement, nor can they be used as a truthful statement. Therefore, we attempted to identify such statements and remove them from the data set. To that end, we used the Google Speech-to-Text API, and provided to it relevant words from our domain. Then, we checked whether the players who played cards not according to the rules stated that they had played illegal cards, and if so, we have removed these samples from the data-set. That method resulted in 3350 samples being removed.

We recruited 156 test subjects who played the game in English using Amazon's Mechanical Turk service [PCI10]. Subjects' demographic information is shown in Table 3.1. We collected a total of 10,788 labeled samples. 7,585 samples were labeled as true (70.3%), and 3,203 samples were labeled as false (29.7%). Each sample lasts about 4 seconds.

TABLE 3.1: Subjects' demographic information

Gender	Male	97
	Female	59
Education level	High-school	62
	Bachelor's	77
	Master's	13
	PhD	4
Average age		35.9

As mentioned, we divided the subjects into two groups: subjects whose data is used only for training and subjects used only for testing. In the training set, we had 111 subjects, and in the test set, we had 45 subjects. It should be noted that in our data set, different recordings are made in different environments, unlike most data sets in which all recordings are gathered from the same environment (i.e., the same microphones). This fact makes our problem more complex, but also more realistic, as, when used in practice, it is anticipated that the data is gathered from many different sources, and each person uses her own recording device. See Table 3.2 for a summary of the data-set division.

TABLE 3.2: Data-set Division

Set	**Subjects**		**Samples**
Train	Male	66	7529
	Female	45	
Test	Male	31	3259
	Female	14	

Chapter 4

Comparative Hint Approach Meta-Learning (CHAML)

We present CHAML, a model that uses a Comparative Hint Approach and Meta-Learning for deception detection. The model consists of three main modules: the embedding module, the core process, and the classifier.

4.1 Embedding Types

In order to classify the audio samples, they must first be vectorized. Therefore, we use embeddings, which encompass the samples' audio features, and feed them to the core process. We consider two different types of embeddings: Five Sound Features and Wav2Vec 2.0.

4.1.1 Five Sound Features

The Five Sound Features is a vector embedding developed by Mansbach et al. [NHA21]. It contains following five features, which are extracted from the audio samples: MFCC, Mel-scale spectrogram, Spectral contrast, Short-time Fourier transform (STFT), and Tonnetz. The vector embedding size is 193. We note that although [NHA21] used Voice Activity Detector (VAD) to trim the silence parts and background noise for all samples, we observed that this practice did not improve the performance of this embedding method. This is likely because the samples are short, and silent segments may contain clues on whether a statement is true or false. Therefore, in this paper, we do not use VAD.

4.1.2 Wav2Vec 2.0

The Wav2Vec 2.0 model introduced by Baevski et al. [AHAM20] is a framework for self-supervised learning of vector representation from speech audio by pretraining on large quantities of audio data developed by Meta AI. The model attempts to recover a randomly masked portion of the encoded audio feature.

The model consists of three main modules. The first module is a feature encoder composed of a 1-d Convolutional neural network encoder, which downsamples the input raw waveform $\mathcal{X}$ to latent speech representation of 25ms each $\mathcal{Z}$ in T time steps. The second module is a contextualized encoder, which consists of several transformer encoder blocks, transforms the latent representations $\mathcal{Z}$ into contextualized representations $\mathcal{C}$. In addition, there is the quantization module, which takes the speech representation $\mathcal{Z}$ and discretizes them into a finite set of quantized representations $\mathcal{Q}$ by matching them with a codebook for selecting the most appropriate

representation of the audio. The objective is to identify these quantized representations of the masked features using the output of the contextualized network C for each masked time step T by using the contrastive loss function. After pretraining on unlabeled audio data, the model can be fine-tuned on labeled data for downstream tasks.

In this study, we used a Wav2Vec 2.0 xlsr-53 model[1] that was fine-tuned on English using the Common Voice data-set [RMK+19] which currently consists of 7,335 hours in 60 languages of transcribed speech. The vector embedding length is 1024. In addition, we fine tune this model to better fit our data, as follows. We take the context representation of our data from the pre-trained models, starting with an average pooling layer that calculates an averaged vector according to the time dimension, and is followed by a fully connected layer with the Tanh activation function. Finally, there is a fully connected layer for the classification task. Since the Wav2Vec 2.0 model was used as a feature extractor, the weights of the features encoder module of the pre-trained model were not changed during the fine-tuning process. This fine-tuning architecture is inspired by [PD21] due to its similarity to our task and for achieving satisfactory results on their tasks. The hyper-parameters used for fine-tuning are represented in Table 4.1.

TABLE 4.1: Hyper-parameters used for fine-tuning

Parameters	
Sample frequency	16k Hz
Learning rate	1e-4
Training epochs	5
Training batch size	12
Gradient accumulation steps	3
Total train batch size	36

4.2 CHAML - Core

In this model, we were inspired by the idea of meta-learning for few-shot learning for deception detection based on speech. We define each person in our data-set as a “meta-learning task". Each task’s support set contains four randomly sampled examples, two per class, and the rest are the query set. In the training phase, each sample was trained with four examples randomly sampled from the training set of the person, i.e., each sample appears as a query, and may appear as an example in the support-set of other samples. Clearly, in the evaluation phase, we do not have different pairs of labeled examples for each query sample, and we use the same four examples for all query samples of a task. As mentioned in Table. 3.2, there are 45 testing subjects (i.e., tasks), and for each of them, we have four labeled examples, two per class. Therefore, in total, we have 90 True and 90 False samples in the support set of the testing tasks. See sample partition details in Table 4.2.

The CHMAL model fixes the order of the provided examples by first using the true examples and then the false examples. For each class’ examples, we calculated its element-wise product; this is known to have the ability to catch similarities or

[1] https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english

TABLE 4.2: Support-Query Division

			Support Set	Query Set	Total Samples
Train	**True**	5288	ALL	ALL	7529
	False	2241			
Test	**True**	2297	90	2207	3259
	False	962	90	872	

discrepancies between the vectors. Each pair is then concatenated with the given product and fed into a few fully connected layers with the ReLU activation function and dropout. The length of the output vector is twice the embedding size. The intuition behind it was to find the relation between the examples of each class and ultimately return its general representation that contains the features that characterize each class. The results from both the positive and negative pairs are combined with the tested sample into a vector that is fed into a neural classifier. Finally, CHAML returns the probability that the tested sample belongs to each of the classes. In the training phase, the weights of the preprocess layers and the neural classifier are all updated after each epoch.

We believe that the model compares the tested sample with the examples of each class in order to find which class is more similar to the tested sample, which helps it to provide a more accurate prediction. An illustration of CHAML's architecture is depicted in Figure 4.1.

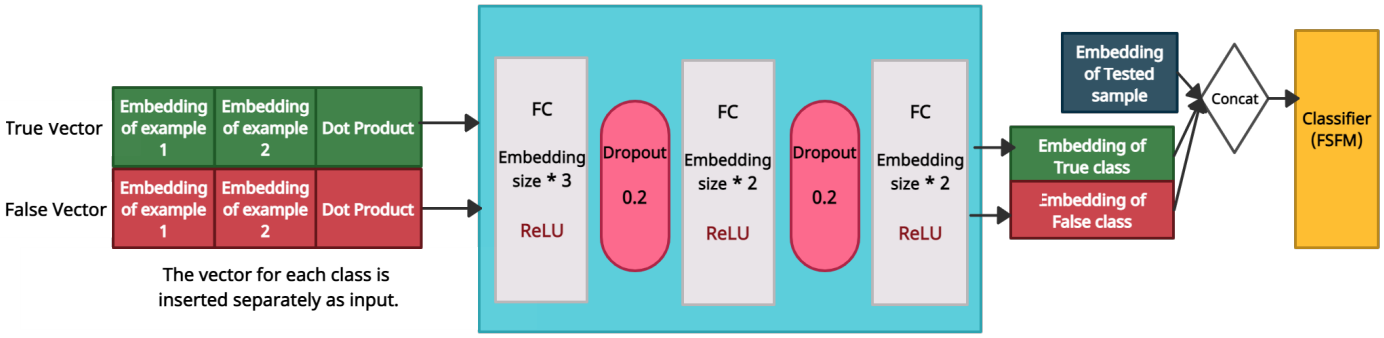

FIGURE 4.1: An illustration of CHAML Model.

4.3 FSFM Classifier

For our model's classification task, we use the Five Sound Feature Model (FSFM) classifier from [NHA21], which achieved the highest scores for the deception detection task in a very similar environment. The FSFM classifier is a multi-layer perceptron network that consists of three fully-connected layers, using the ReLU activation function and dropout after each. The output layer uses a softmax activation function with two classes. The model uses an ADAM optimizer. This model was built especially for the five-sound features embedding mentioned in Section 4.1.1

and achieved good results for our problem; therefore, we selected it as the classifier for the detection task.The model illustration is provided in Figure 4.2.

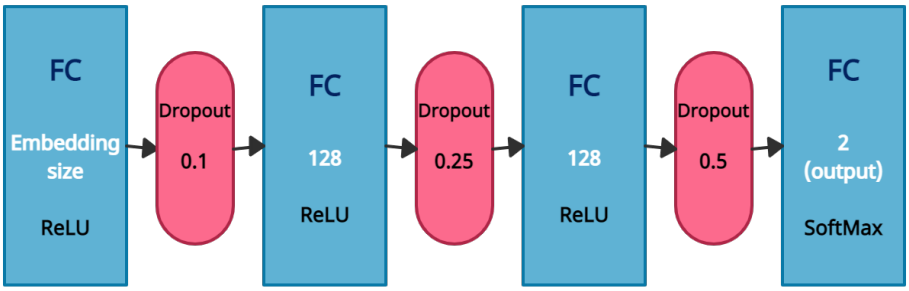

FIGURE 4.2: An Illustration of FSFM Model.

The complete CHAML training process is presented in Algorithm 1. During the test phase, instead of randomly sampling the support set for each query sample, we use the support set provided for each task for all queries of that task.

Algorithm 1: CHAML Training Process

Input : Samples divided to tasks
Output: Predicted labels

1. Embedding Collection:

Use Five Sound Features / Wav2Vec 2.0 / other,
for creating an embedding for each of the samples.

2. CHAML-Core:

Create an empty list $model_{INPUTS}$.

a. foreach *task in train tasks* **do**
`/* create support set for each sample */`
foreach *sample in task* **do**
$T_{support}$:=Randomly sample two True samples from the current task.
$F_{support}$:=Randomly sample two False samples from the current task.
Add ($sample$, $T_{support}$, $F_{support}$) to $model_{INPUTS}$ list.
end
end

b. Shuffle $model_{INPUTS}$ list.

c. Train the model:
`/* item:(`$sample$`,` $T_{support}$`,` $F_{support}$`)` `*/`
for *50 epochs* **do**
foreach *item in* $model_{INPUTS}$ **do** `// batch size = 512`
foreach *class pair in support set* **do** `//` $(s1, s2)$
$pair_{MUL} := s1 * s2$ `// element-wise product`
$pair_{CON} := s1 \cdot s2 \cdot pair_{MUL}$ `// concentration`
Feed $pair_{CON}$ **into model layers:**
$FC\ (ReLU) \rightarrow (embedding\_size * 3)$
$Dropout(0.2)$
$FC\ (ReLU) \rightarrow (embedding\_size * 2)$
$Dropout(0.2)$
$FC\ (ReLU) \rightarrow (embedding\_size * 2)$
Output: $CLASS_{vec}$
`// CLASS = TRUE/FALSE`
end
3. Classifier: `// FSFM`
$sample_{CON} := sample \cdot T_{vec} \cdot F_{vec}$
`// concentration`
Feed $sample_{CON}$ **into classifier layers.**
end
Update all layers parameters (including classifier layers).
end

Chapter 5

Evaluation

5.1 Baseline Methods

We consider four baseline methods, and we compare their performance to that of CHAML. Namely, we consider a method using FSFM as a classifier without fine-tuning, and a method using FSFM that is fine-tuned based on the support sets. In addition, we consider the Prototypical [SSZ17] and MAML [FAL17] networks, which are commonly used for meta-learning.

5.1.1 FSFM - Fine-Tuning

In this method, we trained the FSFM model mentioned in 4.3 on the embedding vectors. During the evaluation phase, for each test task, we first continued training the model on the four examples provided (the support set) and then predicted the value on the query of that test task.

5.1.2 Prototypical Network

A prototypical network [SSZ17] is one of the well-known meta-learning methods and is based on similarity. For each task, the model learns a prototypical embedding, which is a common representation for each class in the support set and matches each query embedding to each class's prototypical embedding to find the most similar class. The prototypical network uses cosine similarity for measuring the similarity between each query embedding and each class prototypical embedding.

5.1.3 Model-Agnostic Meta-Learning (MAML)

Model agnostic meta-learning (MAML) [FAL17] is a meta-learning framework that learns the model parameters that can be fine-tuned very quickly when applied to different tasks. The model is initialized with random weights, and iteratively, for each task in the training tasks, fine-tunes a copy of the primary model. Then the weights of the copy are updated using the loss from the query samples in the task by stochastic gradient descent. At the end of each training epoch, the losses and gradients from all queries are accumulated. MAML then calculates the derivative of the mean loss concerning the primary model's weights, and updates those weights in the primary model. In the evaluation phase, a copy of the primary model is fine-tuned on each support set of the test tasks and evaluated on the query set of the same task.

5.2 Results

In this section, we describe our results for deception detection. All models are trained using 50 epochs and a batch size of 512. In addition, we use weighted categorical cross-entropy loss to deal with the imbalance of our data. The results presented in this paper are the average of 30 different executions, each with a different seed. The scores are calculated on the query sets of the test tasks and can be seen in Table 5.1.

TABLE 5.1: Comparison of models performance

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
FSFM - *Five-features* [NHA21]	55.82	0.2977	0.4122	0.3444
FSFM - *Wav2Vec 2.0*	60.46	0.3411	0.4208	0.3745
FSFM-FT - *Five-features*	55	0.3129	0.4922	0.3822
FSFM-FT - *Wav2Vec 2.0*	54.3	0.3051	0.4803	0.3731
Prototypical [SSZ17] - *Five-features*	53.66	0.294	0.4547	0.3548
Prototypical [SSZ17] - *Wav2Vec 2.0*	52.24	0.2765	0.4253	0.335
MAML [FAL17] - *Five-features*	58.8	0.3148	0.3865	0.3466
MAML [FAL17] - *Wav2Vec 2.0*	61.28	0.349	0.424	0.3823
CHAML - *Five-features*	58.59	0.3345	0.4585	0.3826
CHAML - *Wav2Vec 2.0*	**61.34**	0.3515	0.4307	**0.3857**

As depicted by the Table, the CHAML - Wav2Vec 2.0 model outperformed all other methods with an accuracy of 61.34% and an F1-Score of 0.3857. Table 5.2 provides the confusion matrix for the CHAML model on the Wav2Vec 2.0 embedding.

TABLE 5.2: CHAML - Wav2Vec 2.0 performance on the query set of the test samples

		Predicted	
		True	**False**
Actual	**True**	1513	694
	False	496	376

Next, we compare CHAML to the FSFM model which serves as the classifier in CHAML architecture. As shown in the table, CHAML outperformed the fine-tuning method and improved the results of the FSFM model for both embedding types. We note that the FSFM original work [NHA21] presented higher accuracy and F1-score results than in our experiments. This is due to the fact that in the original work, the entire data set was first shuffled and then split into training and test sets. Therefore, the same person also appeared in the training set and also in the test set. This allowed the model to learn from all types of people. However, in our current work, there is no overlapping between samples of the same person in both train and test sets. Consequently, the performance decreases as the model is trained on some people but tested on others.

In addition, CHAML is compare to prototypical and MAML methods, which are commonly used for meta-learning. The Prototypical network was trained using 50 epochs for each support set and used weighted categorical cross-entropy loss. For obtaining the prototypical embedding for each class, we use the FSFM architecture without the last layer used for the classification task; the prototypical embedding length is 128. MAML was trained on 50 epochs with 15 epochs for each task's fine-tuning and used the FSFM model as the meta-learner. MAML performed better when using random weighted sampling rather than weighted categorical cross-entropy loss for accounting for the imbalance in the data. Clearly, since the Prototypical network and MAML are trained on each task separately, they must use a single support set for all the query samples, unlike CHAML, which has different support sets for each query sample in the training tasks. The results show that CHAML outperforms other meta-learning methods. The Prototypical network achieved the lowest accuracy compared to all other methods and a lower F1-score compared to CHAML, while MAML achieved similar results. The averaged computation times of the three models can be seen in table 5.3. As shown, CHAML is over 25 times faster than MAML while having a similar computation time to the Prototypical network, but achieves much higher performance.

TABLE 5.3: CHAML vs. meta-learning methods Average Computation Times

Model	**Avg. Time in sec.**
MAML [FAL17]	1047.71
Prototypical [SSZ17]	27.22
CHAML	41.07

Moreover, we find that using the Wav2Vec 2.0 embedding 4.1.2 for the deception detection task on our data, seems to have better results in the main models: FSFM, MAML, and CHAML. Interestingly, even in the FSFM model, which was specially designed for the five-sound features embedding, the Wav2Vec 2.0 embedding performs better.

In order to confirm that CHAML uses the examples and does not ignore them, we conduct another experiment in which the support set was mixed and did not have the two true examples first and the two false examples later (as required by CHAML). In this case, the F1-score has significantly decreased from 0.3857 to 0.3707 in the Wav2Vec 2.0 setting and from 0.3826 to 0.332 in the Five-Features setting. This indicates that CHAML relies on the examples for providing an accurate prediction.

Chapter 6

Conclusions & Future work

In this study, we have proposed CHAML, a comparative hint approach meta-learning for deception detection based on speech. The method is based on the meta-learning approach in which a model is trained on various learning tasks to enable it to solve new learning tasks using only a few samples. In our approach, we added some labeled samples (the support set) to each unlabeled sample (the query set) from the same task to predict its label. Our setting differs from the typical meta-learning problem since our data comes from different environments (and different people), whereas in the classical meta-learning framework, data from one environment is divided into different tasks. In addition, in typical meta-learning, there is no overlap between the classes in the training tasks and those in the test task. However, in our approach, classes are the same in all tasks, but the task environment differentiates each one from the other. Therefore, typical meta-learning methods do not perform well in our task, while CHAML, our proposed method, manages to gather relevant information from the support sets and improves the model's performance.

In future work we attempt to find a way to add attention to CHAML, so that it learns how to attend to the relevant parts of each example when making a prediction. We note that our method is not limited to deception detection and can also be applied to other environments, in which all tasks, both in training set and in testing set, include the same classes (or regression problems), but the samples in different tasks highly differ from each other. Examples of such environments include emotion recognition (with many different people; each person is a different task), handwriting recognition (with different people; each person is a different task), and age and sex determination of different animals (each animal is its task), and determining whether sensor data collected from different environments indicates that human intervention is required. In future work, we intend to test CHAML on some of these environments.

Bibliography

[AHAM20] Baevski Alexei, Zhou Henry, Mohamed Abdelrahman, and Auli Michael. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*, 2020.

[BSN20] Dan Barsever, Sameer Singh, and Emre Neftci. Building a better lie detector with bert: The difference between truth and lies. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[C+03] National Research Council et al. *The polygraph and lie detection*. National Academies Press, 2003.

[CR21] Li-Wei Chen and Alexander Rudnicky. Exploring wav2vec 2.0 fine-tuning for improved speech emotion recognition. *arXiv preprint arXiv:2110.06309*, 2021.

[DJ82] Earl F Dulaney Jr. Changes in language behavior as a function of veracity. *Human Communication Research*, 9(1):75–82, 1982.

[DLM+03] Bella M DePaulo, James J Lindsay, Brian E Malone, Laura Muhlenbruck, Kelly Charlton, and Harris Cooper. Cues to deception. *Psychological bulletin*, 129(1):74, 2003.

[DSHW10] Alice Dechêne, Christoph Stahl, Jochim Hansen, and Michaela Wänke. The truth about the truth: A meta-analytic review of the truth effect. *Personality and Social Psychology Review*, 14(2):238–257, 2010.

[DZL+19] Mingyu Ding, An Zhao, Zhiwu Lu, Tao Xiang, and Ji-Rong Wen. Face-focused cross-stream network for deception detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7802–7811, 2019.

[EF03] Paul Ekman and Wallace V Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Ishk, 2003.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[FBC12] Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171–175, 2012.

[FC21] Kexin Feng and Theodora Chaspari. Few-shot learning in emotion recognition of spontaneous speech using a siamese neural network with adaptive sample pair formation. *IEEE Transactions on Affective Computing*, 2021.

[FLC17] Haytham M Fayek, Margaret Lech, and Lawrence Cavedon. Evaluating deep learning architectures for speech emotion recognition. *Neural Networks*, 92:60–68, 2017.

[FTDG17] Noroozi F., Sapiński T., Kamińska D., and Anbarjafari G. Vocal-based emotion recognition using random forests and decision tree. *Int. J. Speech Technol.*, 25:1–8, 2017.

[GAH17] Mandar Gogate, Ahsan Adeel, and Amir Hussain. Deep learning driven multimodal fusion for automated deception detection. In *2017 IEEE symposium series on computational intelligence (SSCI)*, pages 1–6. IEEE, 2017.

[GLF+21] Gaël Guibon, Matthieu Labeau, Hélène Flamein, Luce Lefeuvre, and Chloé Clavel. Few-shot emotion recognition in conversation with sequential prototypical networks. *arXiv preprint arXiv:2109.09366*, 2021.

[GSS+06] M. Graciarena, E. Shriberg, A. Stolcke, F. Enos, J. Hirschberg, and S. Kajarekar. Combining prosodic lexical and cepstral systems for deceptive speech detection. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I, 2006.

[HA15] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[HBB+05] Julia Bell Hirschberg, Stefan Benus, Jason M Brenier, Frank Enos, Sarah Friedman, Sarah Gilman, Cynthia Girand, Martin Graciarena, Andreas Kathol, Laura Michaelis, et al. Distinguishing deceptive from non-deceptive speech. 2005.

[HS21] Andrey Lucas Herchonvicz and Rafael de Santiago. Deep neural network architectures for speech deception detection: A brief survey. In *EPIA Conference on Artificial Intelligence*, pages 301–312. Springer, 2021.

[HYT14] K. Han, D. Yu, and I. Tashev. Speech emotion recognition using deep neural network and extreme learning machine. In *Proceedings of INTERSPEECH ISCA Singapore*, 2014.

[JNB+] M. Jain, S. Narayan, P. Balaji A. Bhowmick, R.K. Muthu, K.P. Bharath, and R. Karthik. Speech emotion recognition using support vector machine.

[KHD06] Mark L. Knapp, Roderick P. Hart, and Harry S. Dennis. An Exploration of Deception as a Communication Construct. *Human Communication Research*, 1(1):15–29, 03 2006.

[KJB+19] Ruhul Amin Khalil, Edward Jones, Mohammad Inayatullah Babar, Tariqullah Jan, Mohammad Haseeb Zafar, and Thamer Alhussain. Speech emotion recognition using deep learning techniques: A review. *IEEE Access*, 7:117327–117345, 2019.

[KMPC18] Gangeshwar Krishnamurthy, Navonil Majumder, Soujanya Poria, and Erik Cambria. A deep learning approach for multimodal deception detection. *arXiv preprint arXiv:1803.00344*, 2018.

[KZS+15] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[MdSD20] Felipe Mateus Marcolla, Rafael de Santiago, and Rudimar LS Dazzi. Novel lie speech classification by using voice stress. In *ICAART (2)*, pages 742–749, 2020.

[MM20] Leena Mathur and Maja J Matarić. Introducing representations of facial affect in automated multimodal deception detection. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, pages 305–314, 2020.

[MMS+22] Merylin Monaro, Stéphanie Maldera, Cristina Scarpazza, Giuseppe Sartori, and Nicolò Navarin. Detecting deception through facial expressions in a dataset of videotaped interviews: A comparison between human judges and machine learning models. *Computers in Human Behavior*, 127:107063, 2022.

[NFS03] T. Nwe, S. Foo, and L. De Silva. Speech emotion recognition using hidden markov models. *Speech Communication*, 41:603–623, 2003.

[NHA21] Mansbach Noa, Neiterman Evgeny Hershkovitch, and Azaria Amos. An agent for competing with humans in a deceptive game based on vocal cues. *Proc. Interspeech 2021*, pages 4134–4138, 2021.

[NOA16] H. Nasri, Wael Ouarda, and A. Alimi. Relidss: Novel lie detection system from speech signal. *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–8, 2016.

[NS22] Anugunj Naman and Chetan Sinha. Fixed-maml for few-shot classification in multilingual speech emotion recognition. In *Machine Intelligence and Smart Systems*, pages 473–483. Springer, 2022.

[OCCH11] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*, 2011.

[PCI10] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.

[PD21] Fivian Pascal and Reiser Dominique. Speech classification using wav2vec 2.0, 2021. https://www.zhaw.ch/storage/engineering/institute-zentren/cai/BA21_Speech_Classification_Reiser_Fivian.pdf.

[PR77] John A Podlesny and David C Raskin. Physiological measures and the detection of deception. *Psychological bulletin*, 84(4):782, 1977.

[PRAMB15] Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, and Mihai Burzo. Deception detection using real-life trial data. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 59–66, 2015.

[PRF21] Leonardo Pepino, Pablo Riera, and Luciana Ferrer. Emotion recognition from speech using wav2vec 2.0 embeddings. *arXiv preprint arXiv:2104.03502*, 2021.

[RMK+19] Ardila Rosana, Branson Megan, Davis Kelly, Henretty Michael, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

[SHD+20] Anastasis Stathopoulos, Ligong Han, Norah Dunbar, Judee K Burgoon, and Dimitris Metaxas. Deception detection in videos using robust facial features. In *Proceedings of the Future Technologies Conference*, pages 668–682. Springer, 2020.

[SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[TO68] Richard I Thackray and Martin T Orne. A comparison of physiological indices in detection oe deception. *Psychophysiology*, 4(3):329–339, 1968.

[TP98] Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.

[TRB+16] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5200–5204. IEEE, 2016.

[Tro38] Paul V Trovillo. History of lie detection. *Am. Inst. Crim. L. & Criminology*, 29:848, 1938.

[VBL+16] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[Vri00] Aldert Vrij. *Detecting lies and deceit: The psychology of lying and implications for professional practice*. Wiley, 2000.

[VVL22] Nik Vaessen and David A. Van Leeuwen. Fine-tuning wav2vec2 for speaker recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7967–7971, 2022.

[WSDS18] Zhe Wu, Bharat Singh, Larry Davis, and V Subrahmanian. Deception detection in videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[XLT+18] Yue Xie, Ruiyu Liang, Huawei Tao, Yue Zhu, and Li Zhao. Convolutional bidirectional long short-term memory for deception detection with acoustic features. *IEEE Access*, 6:76527–76534, 2018.

אוניברסיטת אריאל בשומרון

תקציר

הפקולטה למדעי הטבע
המחלקה למדעי המחשב

זיהוי שקרים מתוך דיבור מבוסס על מטא-למידה

על ידי נעה מנצבך

ישנה חשיבות גדולה לזיהוי שקרים, במיוחד על ידי שימוש ברמזים בדיבור.
בעוד שישנם מחקרים המנסים לזהות שקרים מתוך דיבור, רובם אינם מחלקים את ההקלטות המשמשות לאימון המודל ואלו המשמשות לבחינת המודל לפי האנשים שאמרו אותם או לפי הסביבה בה כל דגימה הוקלטה. דבר זה עלול לגרום לתוצאות זיהוי פחות אמינות.
במחקר זה, אנו נוקטים בגישת מטא-למידה שבה מודל מאומן על מגוון משימות למידה כדי לאפשר לו לפתור משימות למידה חדשות תוך שימוש במספר דוגמאות בודדות בלבד.
בגישה שלנו, אנו מחלקים את ההקלטות לפי האנשים שאמרו אותם (וסביבת ההקלטה), כלומר, חלק מהאנשים משמשים לאימון המודל, ואחרים משמשים לבחינת המודל, מתוך הנחה שקיימות מספר הקלטות מסווגות עבור כל אדם.
אנו מציגים את CHAML, ארכיטקטורה חדשנית המבוססת על למידה עמוקה, המקבלת כקלט הקלטה שאותה רוצים לסווג, ובנוסף לכך ארבע דוגמאות של אותו אדם, כאשר שתיים מהן מסווגות כאמת ושתיים כשקר.
במחקר זה אנו מראים שהשיטה שלנו משיגה תוצאות מדויקות יותר יחסית לשיטות מתקדמות אחרות לזיהוי שקרים מתוך דיבור ובפרט על שיטות המבוססות על מטא-למידה.

אוניברסיטת אריאל בשומרון

הפקולטה למדעי הטבע

זיהוי שקרים מתוך דיבור מבוסס על מטא-למידה

חיבור זה מוגש כחלק מהדרישות לקבלת התואר "מוסמך האוניברסיטה" (M.Sc.)

במחלקה למדעי המחשב

על ידי:

נעה מנצבך

העבודה הוכנה בהדרכתו של פרופ' עמוס עזריה

ט' כסלו תשפ"ג 3.12.2022

אוניברסיטת אריאל בשומרון

הפקולטה למדעי הטבע

זיהוי שקרים מתוך דיבור מבוסס על מטא-למידה

חיבור זה מוגש כחלק מהדרישות לקבלת התואר "מוסמך האוניברסיטה" (M.Sc.)

במחלקה למדעי המחשב

על ידי:

נעה מנצבך

העבודה הוכנה בהדרכתו של פרופ' עמוס עזריה

ט' כסלו תשפ"ג 3.12.2022