Development of an AI Model Capable of Distinguishing the Reason for an Attention-Seeking Touch

ADRIAN JOSAN, University of Twente, The Netherlands

Existing studies in mediated social touch have primarily focused on the effects of touch on users, often neglecting the need for systems to understand user-initiated touches. This research addressed this gap by developing a machine learning model to determine the intention behind attention-seeking touches, specifically differentiating between comforting and warning gestures. Utilizing insights from prior studies, the proposed model enhanced the understanding of social touch in human-machine interactions. The methodology included a literature review and an experiment with human participants. Participants performed attention-seeking touches on a mannequin arm equipped with a Touch Sensitive Patch (TSP). Data on touch location, intensity, and duration were collected and anonymized. A Random Forest Classifier was primarily used to train the model, with an additional classifier explored. This research demonstrated the potential of machine learning to interpret reason-dependent attention-seeking touch signals, advancing the understanding of social touch.

Additional Key Words and Phrases: Affective-Computing, Social Touch, Attention-Seeking Touch, Mediated Touch

1 INTRODUCTION

In the field of mediated social touch, studies have primarily focused on generating and assessing the effects of touch on users. However, for interactive applications, systems must also possess the capability to comprehend social touches initiated by users. Understanding the intricacies of social touch is fundamental to developing machines with emotional intelligence, particularly in interpreting the purpose behind attention-seeking touch gestures. Existing research states the challenge of accurately mapping touch to emotional states or intentions [6]. To address this, the development of a machine learning model capable of distinguishing the underlying intention of attention-seeking touch holds significant promise. Such a model not only enables robots to respond authentically to touch gestures but also has implications for enhancing mediated touch experiences over distance. Current haptic devices are limited by sensor capabilities, hindering their ability to transmit accurately specific touch sensations. The intention of the attention-seeking touch could be determined through this model and recreated for the person receiving the touch by enabling their haptic actuators in a manner that could potentially be more effective than one-to-one mediation [2, 14]. By developing an AI model capable of distinguishing between comforting and warning attention-seeking touch, this research aims to determine the feasibility of distinguishing the reason for an attention-seeking touch using machine learning. Despite recent advancements in machine learning models capable of discerning various types of touches on robotic skins, a significant gap remains in understanding the specific intention behind each touch. A key insight from existing research, such as Yohanan and MacLean's work on the Haptic Creature [16], emphasizes the substantial advantages of interpreting touch data at a higher level of abstraction. While current models may focus on identifying the type of touch (e.g., a squeeze or a stroke), they fall short in discerning the underlying emotional intent, which is critical for authentic human-robot interactions. By categorizing touches based on intent rather than mere physical characteristics, a robot can better understand and respond to human intentions and messages. For instance, distinguishing between a comforting touch meant to soothe and a warning touch meant to alert can enable a robot to choose appropriate responses, thus enhancing its role as an empathetic and responsive companion. This level of understanding goes beyond simply recognizing touch patterns; it involves interpreting the human's emotional state and intentions, making interactions more meaningful and effective. This study aims to address this gap in knowledge by focusing on the development of a machine learning model trained to distinguish between comforting and warning attention-seeking touches. Participants will be engaged in a series of interactions with a mannequin arm equipped with a Touch Sensitive Patch (TSP) covering it, across diverse scenarios. In some scenarios, participants will be instructed to convey warning messages through touch, while in others, their task will be to provide comfort. The readings from the TSP will be analyzed to capture both the location and intensity of the touch, forming the basis for training the machine learning model. Through this research, the goal is to contribute to the understanding of social touch and explore the potential for creating more authentic human-machine interactions. Ultimately, the research seeks to determine whether attention-seeking touch carries distinct enough cues, independent of context, to differentiate between comforting and warning attention-seeking touches.

1.1 Research Questions

This research tries to answer the following main question:

- Can an AI model be trained that is capable of distinguishing between an attention-seeking touch on the arm with the purpose of warning and comforting?

The main research question is answered by providing answers to the following sub-questions:

(1) What is the current state of the art in ML models for distinguishing between different reasons for touch, particularly social touch?
(2) How can an experiment be designed to effectively collect relevant data from study participants for training the ML model?
(3) Which ML model is suitable for training with data from the Touch Sensitive Patch (TSP) to differentiate between

TScIT 41, July 5, 2024, Enschede, The Netherlands

attention-seeking touches intended for warning and comforting purposes?

1.2 Article Structure

The rest of the research paper is organized as follows:

Section 2 provides a comprehensive review of the current state of the art in machine learning models for distinguishing between different types of social touch, highlighting the existing challenges and limitations.

Section 3 describes the experimental design for data collection, including the setup with the Touch Sensitive Patch (TSP), software tools, participant selection, and the specific choice of scenarios used to elicit comforting and warning touches.

Section 4 explains the methodology for data analysis and model training, detailing the selection of the machine learning algorithm, feature extraction, and evaluation metrics used to assess model performance.

Section 5 presents the results of the study, analyzing the model's accuracy and ability to differentiate between comforting and warning touches, along with any notable findings from the experiment.

Section 6 discusses the implications of the results, potential applications, and limitations of the study, offering insights into future research directions.

Finally, Section 7 concludes the paper by summarizing the key findings and contributions, and suggesting practical implementations for enhancing mediated touch experiences in human-machine interactions.

2 STATE OF THE ART

To gather related literature, Google Scholar and Research Rabbit has been utilized. Using the search terms "social touch", "human-computer interaction", and "models to classify social touch", several research papers have been found that were relevant to the topic of machine learning models capable of distinguishing between reasons for touch. Existing research in the field of mediated social touch has primarily focused on understanding the physiological and psychological effects of touch [10], as well as developing models to recognize and interpret social touch signals. Van Erp and Toet (2015) emphasize the importance of mediated affective touch in modulating physiological responses, fostering trust and affection, and enabling pro-social behavior [14]. However, while significant progress has been made in developing models to classify different types of touch, such as discriminative touch [5, 6, 15], fewer studies have focused on distinguishing the intention or reason behind the touch. Notably, studies using the CoST dataset [7] have achieved high accuracy in touch type detection, but they do not address the distinction between the intent of the touch, such as warning or comforting.

Jones and Yarbrough's framework categorizes touch intent into five main categories: protective, comforting, restful, affectionate, and playful. Specific touch gestures and physical properties such as duration, intensity, and pressure characterize each category [16]. For example, comforting touches often involve sustained gestures like hugging or repetitive actions like stroking, which are distinct in their pressure and duration compared to other intents. Since different emotions can manifest through similar physical gestures, interpreting the intention behind a touch could potentially be more practical than decoding specific emotions.

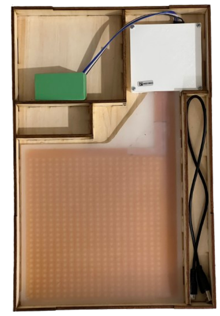

Fig. 1. Touch Sensitive Patch inside casing

Silvera-Tawil et al. (2014) have demonstrated the feasibility of classifying social touch messages using classifiers, providing valuable insights into the potential of AI models in this domain [11]. However, the study relied on instructed touch gestures rather than naturally occurring interactions, which according to Saarinen et al. (2021c) is not ideal since it relies heavily on participants' ability to spontaneously generate social touches. In contrast, this study seeks to address these gaps by using various scenarios to generate natural attention-seeking touches with the specific intention of warning or comforting.

3 DESIGN OF DATA COLLECTION EXPERIMENT

This section describes the design and implementation of the experiment for collecting attention-seeking touch data. Initially, the tools developed and utilized for the experiment are described. Subsequently, the detailed procedures followed to conduct the experiment in an unbiased manner are presented.

3.1 Apparatus/Materials

3.1.1 Touch Sensitive Patch. The sensor utilized for recording touch events is the Touch Sensitive Patch (TSP), developed by Gwenn Englebienne, Henk Waayer, Richard Bults, and Alfred de Vries from the Human-Machine Interaction (HMI) department at the University of Twente (Figure 1). The TSP comprises a 27x19 matrix of sensors embedded within a silicon patch, operating at a refresh rate of 10 Hz. Each sensor in the array provides a touch pressure reading ranging from 0 to 255. The TSP is designed to emulate the elasticity, flexibility, and color of human skin, with a lighter shade for visual representation.

3.1.2 TSP-equipped Mannequin Arm. To provide a more natural and intuitive touch environment, the TSP was mounted on a mannequin arm (Fig. 2). The TSP, made of silicone, was easily wrapped around the forearm region where the attention-seeking touch would occur. Given its thin profile (less than 1 cm in thickness), the final thickness of the forearm remained realistic. To make the TSP less noticeable, the arm was covered with a cotton sleeve, which, after testing, was found to still provide accurate readings.

Fig. 2. TSP-equipped mannequin arm without (left) and with a sleeve (right)

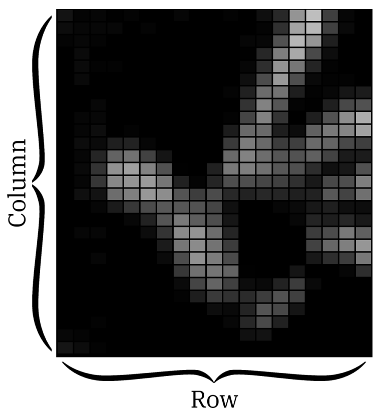

Fig. 3. Frame capturing a grab gesture

3.1.3 Touch Recording Program. To store the touch data recorded by the touch-sensitive patch (TSP), a Python program was developed. It works by saving the pressure values for each sensor on the vertical and horizontal array at the current point in time. The resulting data structure is called a frame and is illustrated in Fig. 3. The program processes each frame by assigning the corresponding timestamp and then compiles the data into a JSON file along with relevant metadata. This metadata is subsequently utilized for labeling during the model training phase. During recording, each frame is shown in real-time on the screen for the researcher to validate the quality of the frames. To ensure a higher consistency in the recording start time, the recording is initiated when at least one sensor detects a threshold value of 30 or more (out of 255). The threshold value of 30 was determined through empirical testing, as it proved to be the most reliable for distinguishing actual touches from random noise. Additionally, a buffer of 2 seconds of pre-threshold recording is maintained to ensure that no touch information prior to the threshold detection is lost.

3.1.4 Scenarios. To generate the attention-seeking touches necessary for training the model to distinguish between warning and comforting touches, four different scenarios were created. Two comforting scenarios involved a person who had either sustained a minor injury after falling off a bike outdoors or was in distress and crying after a phone call inside a university hall. At the end of these scenarios, participants were invited to give a comforting touch to the subject of the scenario to provide support. The first warning scenario involved a person in imminent danger of being hit by a vehicle while walking on the street, where participants were asked to touch them on the forearm to warn them of the danger. The second warning scenario depicted a person blocking a narrow passage in a bar, with participants asked to touch them to warn the subject to move so that others could pass. The pairs of scenarios were written to be different enough from each other while having a similar reason for performing the attention-seeking touch, either comforting or warning.

Fig. 4. Experiment setup

3.1.5 Web Application and Database. To automate the process of survey data collection and the presentation of warning and comforting scenarios to participants, a web application was developed. The website asks the participant to complete a short survey, after which the sequence of scenarios is generated and stored in a database. The sequences are generated to ensure equal representation of scenarios across all participants. Subsequently, the touch recording program automatically assigns the correct metadata for each participant's touch by fetching the corresponding sequence of scenarios from the database. The web application was rendered on a tablet because, compared to a laptop or desktop computer, it occupied less space on the desk where the TSP-equipped arm was placed, thereby minimizing distractions.

3.2 Experiment Procedure

All experiments were performed in a private room where a researcher was present to instruct the participant and ensure the recordings went smoothly. The TSP-equipped mannequin arm was positioned with the hand toward the seated participant to allow them to reach it more easily (Fig. 4).

Upon entering the room where the study took place, participants were welcomed and asked to read and sign a consent form. Following this, each participant received instructions on the experimental procedure. They were informed about the four scenarios they would need to read and decide whether to touch the mannequin arm accordingly. Participants were instructed to focus on making the touch as authentic as possible, without concern for the quality, intensity, or duration of the touch, and without considering the research purpose of the sensor. Participants were asked to imagine that the mannequin arm represented the arm of the person described in the scenario. They were also given the option to abstain from touching the arm if they felt uncomfortable doing so for any scenario. Each participant was then assigned a participant ID, which was used to generate the sequence of scenarios. After each scenario, participants placed the tablet on the desk and performed the touch. The recording started automatically when the participant began touching the arm, and the researcher manually stopped it once the participant removed their hand. The entire procedure took approximately 10 minutes per participant. After the study, participants were thanked for their participation.

This study was approved by the ethics committee of the Faculty of Electrical Engineering, Mathematics, and Computer Science of the University of Twente.

4 DATA ANALYSIS AND MODEL TRAINING

This section describes the steps taken to convert the raw data obtained during experimentation into a suitable format for touch classification. Additionally, the classification algorithm is introduced.

4.1 Data Filtering

Data filtering is essential to ensure the suitability of the data for training a model. The first step in filtering was applied during the recording process. The touch-recording program initiated the recording two seconds before touch detection, ensuring no unnecessary frames were recorded, which could negatively impact the classifier's accuracy. After data was collected, using a Python-based visualization tool, each recording was examined visually. With the help of a filtering tool, any extraneous data at the end of the recordings was removed, ensuring that the last recorded frame was within two seconds after the touch had ended.

4.2 Data Format

Each touch recording consisted of an ordered list of frames labeled with the corresponding touch reason. Each frame had a resolution of 27x19, thus 513 values per frame (Fig. 3), each being an integer value between 0 and 255. Each second of recording resulted in 10 frames.

4.3 Model Selection

Several models were considered for classifying the reason for an attention-seeking touch. However, a Random Forest model (RF) has been selected due to its solid performance in classifying touch gestures by training on the HAART, CoST, and original datasets by previous researchers [1, 4, 12]. While the goal of the model trained in this research is different since it focuses on the reason for touch instead of specific gestures, the nature of the data and the distinct features of touch are similar. Also, the aforementioned papers provide guidance in the process of optimizing the model's performance.

The core concept of this algorithm is to create a forest by training and integrating various types of decision trees, with the final classification result determined by a majority vote among these trees. This method incorporates the 'bagging' technique along with the random selection of features [13].

The Random Forest classifier has been implemented using the Scikit-learn library [9]

4.4 Machine Learning Approach

The sliding window technique was implemented to analyze touch recordings by dividing the entire set of frames into overlapping windows and independently extracting features from each window. This technique helps preserve the relationship between adjacent frames. A window size of 30 frames was selected for the final model as it showed the best balance between accuracy on the test dataset and cross-validation accuracy on the training dataset.

An important factor in improving the performance of the Random Forest model is identifying the optimal set of hyperparameters for the specific use case. Since the best parameters are not known in advance, it is necessary to try all possible values and test their performance against each other. For this purpose, the GridSearchCV function from the Scikit-learn library was utilized. A dictionary (or grid) of sensible hyperparameter values was defined (Table 1). GridSearchCV exhaustively evaluates all possible combinations of hyperparameters provided in the dictionary and assesses each combination using the best average accuracy score after performing 3-fold cross-validation on the training dataset. The best-calculated and final hyperparameters for the model are presented in Table 2. The Random Forest model consists of 100 trees. Each tree has no limit on its maximum depth, requires at least 10 samples to split an internal node, and requires at least 2 samples at each leaf node.

Table 1. Grid of Hyperparameter Values for Hyperparameter Optimization

Hyperparameter	Values
n_estimators	50, 100, 200
max_depth	None, 10, 20, 30
min_samples_split	2, 5, 10
min_samples_leaf	1, 2, 4

Table 2. GridSearchCV-computed Hyperparameters

Hyperparameter	Value
n_estimators	100
max_depth	None
min_samples_split	10
min_samples_leaf	2

The training and testing frames were separated based on the recording to which they belonged, ensuring that no frames from the same recording appeared in both the training and test datasets, thereby preventing biased performance results.

K-fold cross-validation was used to validate the performance of the system on unforeseen data. With this method, the data set is partitioned into k subsets. One subset is stored for testing and the remaining k-*1* subsets are used for training the model. The splitting of data is performed k times such that each subset is tested on. A relatively low k value of 5 was selected to ensure that each train/test group of data samples is sufficiently large to statistically represent the entire data set.

4.4.1 Features set. In machine learning algorithms, recognition relies on the integration of measurable properties of the dataset, known as "features," to distinguish between different data categories. The performance of classification heavily depends on strong features that can differentiate between the reason for touch. The set of features used to train the touch reason classifier was mainly based on the research performed by Viet-Cuong Ta et al. (2015), where a list of features was selected based on previous works that proved effective for extracting distinct features from touch gestures. The extraction of the following three groups of features has been implemented:

Global features represent the general statistics of all the frames from a window of frames:

- Average pressure on 513 channels over all frames
- Maximum value of pressure of all channels over all frames
- Mean pressure over all frames of each column
- Mean pressure over all frames of each row

Channel-based features compute various statistics for each channel, representing a sensor or pixel location on the TSP over all frames of a window. These features provide the spatial relationship between each channel:

- Average pressure of each channel over all frames. The absolute difference between each pair of consecutive frames is computed. Then, the mean of all the absolute differences for each channel over the window length is extracted.
- Percentage of time when a channel has pressure more than a threshold T. T is calculated by taking the 90$^{\text{th}}$ percentile of all pressures of all frames of a window. The percentage is taken by dividing the number of frames in which the channel had a threshold-exceeding pressure by the window size. The result is the portion of time each channel has detected strong pressure.

Sequence features have been extracted using two algorithms that are commonly used for extracting features in time series data. For the classifier, these algorithms will extract the changes of all channels over time:

- Fast Fourier Transform-based features are calculated by applying the Fast Fourier Transform [3] to the average pressures for each frame in a window. The operator takes the highest 15 values.
- Discrete Cosine Transform-based features are calculated by applying the Discrete Cosine transform [8] to the average pressures for each frame in a window.

The relevance of the set of features was later evaluated by measuring their impact on the classifier's performance, which will be discussed in the "Results" section.

5 RESULTS

5.1 Experiment Results

A total of 31 participants took part in the experiment: 24 males and 7 females aged between 18 and 34 years old. Volunteers originated from various countries: The Netherlands (12), Moldova (5), Thailand (2), China (3), Romania (1), France (1), India (1), Saudi Arabia (1), Italy (1), Uzbekistan (1), Poland (1), Taiwan (1), South Korea (1). Each participant read 4 scenarios and was asked to perform an attention-seeking touch corresponding to the scenario: 2 for warning purposes and 2 for comforting purposes. Out of the 124 potential touches, 108 were performed. Participants chose to abstain from attention-seeking touches for 16 scenarios. The majority of touches lasted less than 15 seconds. However, there were some outliers, as three participants performed the touch for longer than 15 seconds for 10 recordings in total. The mean pressure value of all sensors during the recordings is 28.05, the median pressure is 4.5, and the standard deviation of touch pressure is 47.65.

After visually inspecting each recording, distinct patterns in touch gestures were observed. While the majority of participants performed a single touch with the entire surface of their palm, 4 participants used a stroking gesture for comforting touches, 3 participants used a poking gesture for warning touches, and 2 participants used the back of their fingers for warning touches.

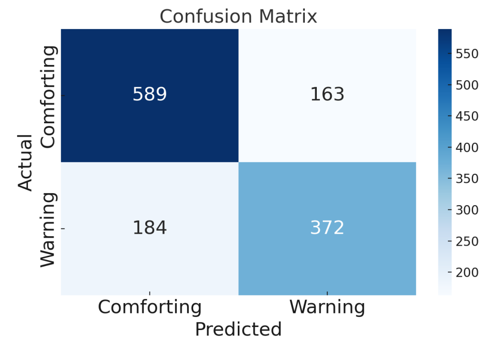

Fig. 5. Classifier's confusion matrix

5.2 Model Training Results

The performance metrics of the Random Forest classifier can be found in Table 3. The confusion matrix representing the prediction summary is depicted in Fig. 5. The accuracy of the model on the testing dataset is 75.23%. The mean cross-vector accuracy on the training dataset is 68.61%. The 95% confidence interval for the cross-validated accuracy is 68.61% ± 9.63%. The comparison of the model's performance with different subsets of the feature set is shown in Table 4. The results confirm that extracting all of the features described in Section 4.4.1 results in the best performance of the model. The performance difference between "Global" features and "Global + Channel-based" features is negligible. However, there is a 7.42% increase in accuracy when extracting the entire feature set compared to the "Global" set of features alone.

5.3 Model Comparison

A CNN Long Short-Term Memory Network model has also been trained on the dataset for comparison purposes. The CNN-LSTM model processes sequences of recording frames by first applying a series of convolutional filters to each frame to extract spatial

Table 3. Performance metrics of the model

Metric	**Score**
Accuracy	0.7523
Precision	0.7624
Recall	0.6061
F1 Score	0.6754
ROC AUC	0.8357
Mean CV Accuracy	0.6861

Table 4. Incremental feature set model performance.

Features	Acc. on Train	Acc. on Test
Global	65.79%	67.81%
Global + Channel-based	64.94%	66.90%
Global + Channel-based + Sequence	68.61%	75.23%

Table 5. Comparison of performance metrics between the Random Forest model and the CNN-LSTM model.

Metrics	Random Forest Model	CNN-LSTM Model
Accuracy	73.47%	61.90%
Precision	69.53%	56.00%
Recall	66.91%	62.00%
F1 Score	68.19%	57.00%

features. Each frame is convolved with 32 filters of size 3x3, resulting in a feature map. These maps are then down-sampled using max pooling. The flattened feature maps from each frame are then passed through an LSTM layer with 64 memory units to capture temporal dependencies across the sequence of frames. Finally, a dense layer with a sigmoid activation function outputs a single probability value, indicating whether the input sequence belongs to the "Warning" or "Comforting" class.

The relative results are in Table 5. The model's accuracy on the training data is 61.9%, which is 11.57% lower than the RF model's performance.

6 DISCUSSION

We attempted to train an AI model to differentiate between comforting and warning attention-seeking touches on the arm. The performance results of the final model suggest that it is feasible to distinguish the reasons for attention-seeking touches. With an accuracy of 75.23%, the Random Forest classifier outperforms random guessing in correctly classifying the reason. However, the model's accuracy is still not high for a binary classifier, indicating that it is not yet suitable for real-world application. Notably, achieving an accuracy higher than 50% demonstrates that attention-seeking touches for different reasons have sufficiently distinct features to be classifiable by the model.

After analyzing the touch recordings, it has been noted that different individuals employ different types of gestures for attention-seeking touches with the same reason. For instance, while most participants chose to grab the subject's arm as a warning gesture (Fig. 3), some opted to poke it or push it with the back of their fingers (Fig. 6). This variation is not problematic for a model distinguishing between warning and comforting touches, as no participant used poking for comforting touches. However, it could pose a challenge for discerning touches meant for rejection, which might also involve a poking gesture. Nonetheless, distinguishing features such as the duration and applied pressure of a poking gesture might vary depending on the reason. Future researchers could expand the classes of reasons for touch to further investigate this aspect.

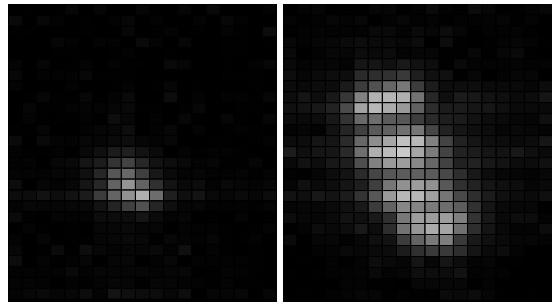

Fig. 6. Frames capturing a poking gesture (left image) and a push with the back of the fingers (right image)

Given the promising performance of the Random Forest algorithm, most efforts were focused on optimizing its performance. In contrast, the CNN Long Short-Term Memory Network (CNN-LSTM) model showed poorer performance in our tests, with accuracy scores only slightly better than random guessing (Table 5). However, it is not excluded that with careful model tuning, the CNN-LSTM model could reach or even exceed the scores of the Random Forest.

The increase in performance from including global, channel-based, and sequence-based features from the touch recordings demonstrates that factors such as location, contact size, pressure, and temporal variations all contribute to identifying the reason for an attention-seeking touch. This set of features can be further validated and extended with new features that aid in distinguishing touch patterns specific to each reason.

Another notable observation is that the mean cross-validation accuracy of the Random Forest model is 68.61%, compared to 75.23% accuracy on the test data set. The 6.62% difference suggests that the current model is not as robust on unseen data, indicating slight overfitting on the training data set.

7 CONCLUSION

This research aimed to develop an AI model capable of distinguishing between comforting and warning attention-seeking touches on the arm. The Random Forest classifier demonstrated promising performance with an accuracy of 75.23%, indicating that it is feasible to differentiate the reasons behind attention-seeking touches based on features such as location, contact size, pressure, and temporal variations. Although the model is not yet suitable for real-world applications due to its current accuracy level, it lays the groundwork for further exploration and refinement in the field of human-machine interaction.

Existing machine learning models have primarily focused on classifying touch types rather than the intention behind touches. Studies using the CoST dataset have achieved high accuracy in touch type detection but do not address the distinction between touch intents such as warning or comforting. This research contributes to the state of the art by demonstrating that attention-seeking touches for different reasons carry distinct enough features to be classifiable by a model.

The experiment was designed to collect touch data using a Touch Sensitive Patch (TSP) mounted on a mannequin arm. Participants were presented with four scenarios—two for warning and two for comforting—and were instructed to perform the corresponding touches. The data, including touch location, intensity, and duration, were recorded and used to train the model. The experiment ensured that each touch was as authentic as possible by instructing participants to focus on the intent rather than the quality, intensity, or duration of the touch.

The Random Forest model was chosen for its robust performance in classifying touch gestures in previous research. Our implementation of the Random Forest model yielded a solid accuracy of 75.23% on the test set. This suggests that the model can effectively utilize the extracted features to distinguish between warning and comforting touches. Despite its promising results, there is still room for improvement. Future research should focus on refining the feature set and exploring additional features that may further improve classification accuracy.

Future work should also investigate the application of more advanced models, such as deep learning techniques, to enhance performance. Moreover, expanding the classes of reasons for touch and incorporating more diverse scenarios could provide a more comprehensive understanding of social touch.

ACKNOWLEDGEMENTS

We would like to thank Angelika Mader and Gwenn Englebienne, for their guidance and support throughout this research. We are also grateful to the Human-Machine Interaction (HMI) department at the University of Twente for providing the necessary resources and facilities. Finally, we thank all the participants who took part in the experiments for their time and cooperation.

REFERENCES

[1] Yona Falinie Abdul Gaus, Temitayo Olugbade, Asim Jan, Rui Qin, Jingxin Liu, Fan Zhang, Hongying Meng, and Nadia Bianchi-Berthouze. 2015. Social Touch Gesture Recognition using Random Forest and Boosting on Distinct Feature Sets. https://doi.org/10.1145/2818346.2830599

[2] Heather Culbertson, Cara M. Nunez, Ali Israr, Frances Lau, Freddy Abnousi, and Allison M. Okamura. 2018. A social haptic device to create continuous lateral motion using sequential normal indentation. In *2018 IEEE Haptics Symposium (HAPTICS)*. 32–39. https://doi.org/10.1109/HAPTICS.2018.8357149

[3] P. Duhamel and M. Vetterli. 1990. Fast fourier transforms: A tutorial review and a state of the art. *Signal Processing* 19, 4 (April 1990), 259–299. https://doi.org/10.1016/0165-1684(90)90158-U

[4] Anna Flagg and Karon Maclean. 2013. Affective touch gesture recognition for a furry zoomorphic machine. 25–32. https://doi.org/10.1145/2460625.2460629

[5] Merel M. Jung, Xi Laura Cang, Mannes Poel, and Karon E. MacLean. 2015. Touch Challenge '15: Recognizing Social Touch Gestures. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (Seattle, Washington, USA) *(ICMI '15)*. Association for Computing Machinery, New York, NY, USA, 387–390. https://doi.org/10.1145/2818346.2829993

[6] Merel M. Jung, Mannes Poel, Ronald Poppe, and Dirk Heylen. 2017. Automatic recognition of touch gestures in the corpus of social touch. 11, 1 (Mar 2017). https://doi.org/10.1007/S12193-016-0232-9

[7] Merel M. Jung, Ronald Poppe, Mannes Poel, and Dirk K.J. Heylen. 2014. Touching the Void – Introducing CoST: Corpus of Social Touch. In *Proceedings of the 16th International Conference on Multimodal Interaction* (Istanbul, Turkey) *(ICMI '14)*. Association for Computing Machinery, New York, NY, USA, 120–127. https://doi.org/10.1145/2663204.2663242

[8] J. Makhoul. 1980. A fast cosine transform in one and two dimensions. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, 1 (Feb. 1980), 27–34. https://doi.org/10.1109/TASSP.1980.1163351

[9] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. 2012. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (Jan. 2012).

[10] Aino Saarinen, Ville Harjunen, Inga Jasinskaja-Lahti, Iiro P. Jääskeläinen, and Niklas Ravaja. 2021. Social touch experience in different contexts: A review. *Neuroscience Biobehavioral Reviews* 131 (2021), 360–372. https://doi.org/10.1016/j.neubiorev.2021.09.027

[11] David Silvera-Tawil, David Rye, and Mari Velonaki. 2014. Interpretation of Social Touch on an Artificial Arm Covered with an EIT-based Sensitive Skin. *International Journal of Social Robotics* 6, 4 (Nov. 2014), 489–505. https://doi.org/10.1007/s12369-013-0223-x

[12] Viet-Cuong Ta, Wafa Johal, Maxime Portaz, Eric Castelli, and Dominique Vaufreydaz. 2015. The Grenoble System for the Social Touch Challenge at ICMI 2015. 391–398. https://doi.org/10.1145/2818346.2830598

[13] Michel Valstar, Björn Schuller, Kirsty Smith, Florian Eyben, Bihan Jiang, Sanjay Bilakhia, Sebastian Schnieder, Roddy Cowie, and Maja Pantic. 2013. AVEC 2013: the continuous audio/visual emotion and depression recognition challenge. In *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge (AVEC '13)*. Association for Computing Machinery, New York, NY, USA, 3–10. https://doi.org/10.1145/2512530.2512533

[14] Jan B. F. van Erp and Alexander Toet. 2015. Social Touch in Human–Computer Interaction. *Frontiers in Digital Humanities* 2 (2015). https://doi.org/10.3389/fdigh.2015.00002

[15] Siewart van Wingerden, Tobias J. Uebbing, Merel M. Jung, and Mannes Poel. 2014. A Neural Network Based Approach to Social Touch Classification. In *Proceedings of the 2014 Workshop on Emotion Representation and Modelling in Human-Computer-Interaction-Systems* (Istanbul, Turkey) *(ERM4HCI '14)*. Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/2668056.2668060

[16] Steve Yohanan and Karon E. MacLean. 2012. The Role of Affective Touch in Human-Robot Interaction: Human Intent and Expectations in Touching the Haptic Creature. *International Journal of Social Robotics* 4, 2 (April 2012), 163–180. https://doi.org/10.1007/s12369-011-0126-7

A APPENDIX A

During the preparation of this work, the author used OpenAI's ChatGPT 3.5[1] in order to generate the four scenarios for the user study of this research and Grammarly[2] to improve the readability of the work. This helped to formulate nuanced and clear scenarios. After using these tools/services, the author reviewed and edited the content as needed and takes full responsibility for the content of the work. The author edited the last lines of the retrieved scenarios to clearly state the goal of the touch.

[1] https://chatgpt.com/
[2] http://grammarly.com/