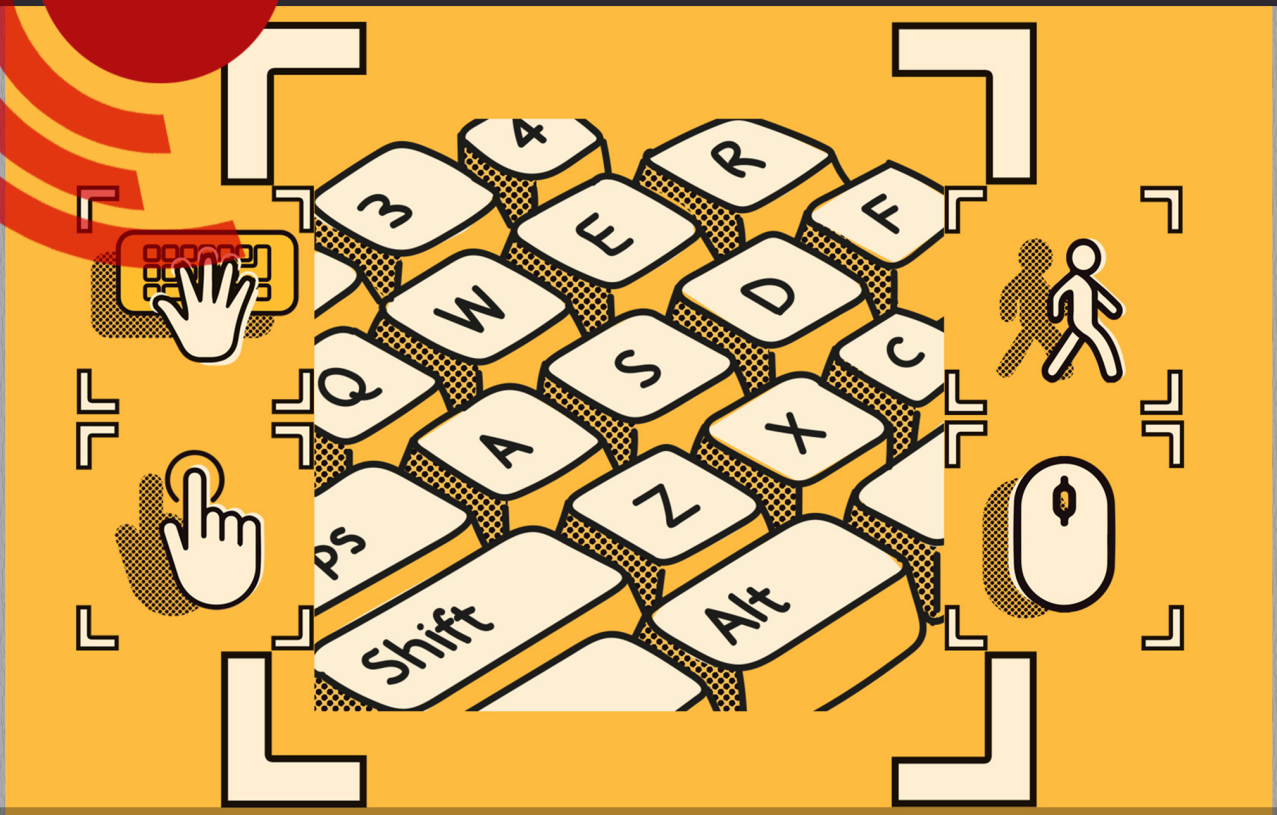
CREST Centre for Research and Evidence on Security Threats



Collecting and Leveraging Identity Cues with Keystroke Analysis (CLICKA)

FULL REPORT

SEPTEMBER 2021

Oliver Buckley

SEPTEMBER 2021

Collecting and Leveraging Identity Cues with Keystroke Analysis (CLICKA)

FULL REPORT

Oliver Buckley
School of Computing Sciences, University of East Anglia

This report details the key findings of work conducted by the CREST commissioned project *Collecting And Leveraging Identity Cues With Keystroke Analysis (CLICKA)*. You can view all the outputs from this project at: crestresearch.ac.uk/projects/clicka/

This research was funded by the Centre for Research and Evidence on Security Threats – an independent Centre commissioned by the Economic and Social Research Council (ESRC Award: ES/N009614/1) and which is funded in part by the UK security and intelligence agencies and Home Office.

www.crestresearch.ac.uk

TABLE OF CONTENTS

EXECUTIVE SUMMARY

BACKGROUND

Keystroke dynamics is the analysis of how an individual uses a keyboard. These typing behaviours can be as uniquely identifiable as a person's handwriting or signature and this data can reveal identifying characteristics about an individual.

The analysis of the typing rhythm and cadence of a user can be used to identify an individual, while also providing information about the person sitting at the keyboard, which can include characteristics such as handedness, hand size, or typing style.

Whereas previous work largely focused on confirming the identity of an anonymous user, this work aimed to understand more about the individual using the device by determining the name and native language of an anonymous user, based on how they type.

The first experiment focused on determining the name of an anonymous user and collected typing samples from 84 users. The participants completed several typing exercises where the timing of each keystroke was recorded. The typing data were subdivided into substrings of two characters (bigrams), and the time between releasing the first key and pressing the second (the flight time) was calculated. The research hypothesis was that those bigrams with a greater familiarity with the user will have a discernibly higher ranking than those that are not as commonly used.

The research made use of machine learning classifiers to develop a model that is capable of a balanced accuracy prediction of approximately 70% of the bigrams in an anonymous user's name.

The second experiment aimed to predict the native language of an individual based on an analysis of their typing behaviours. The experiment collected data from 492 participants across five native languages (English, French, German, Spanish, and Italian) with around 100 people in each group. The participants were again required to complete typing exercises and this data was segmented into bigrams.

Again, machine learning classifiers were used to predict a user's native language. In the first instance, the research aimed to distinguish whether English was the user's native language (i.e. English versus French, Spanish, German, Italian) with a balanced accuracy of 71%, using the SVC classifier.

When predicting the native language of an individual, based on five languages, the approach achieved a balanced accuracy of 45%. While this offers significant room for improvement it does perform notably better than a random prediction.

The research established that users display repeatable and predictable typing behaviours based on familiar identity or linguistic data.

INTRODUCTION

The traditional approach to securing a computer, device, or service will typically rely on the use of a username and password pair. However, this approach is far from perfect and suffers from obvious challenges. Users are required to create credentials that are both memorable to the user but also not easily guessed or inferred by a malicious third party. These two criteria are incongruous and as a result, the instances of compromised accounts are all too common. The involvement of a human in the process means that it will often be difficult to create security credentials that conform to security guidelines for generating strong passwords *[1].*

Once a user has been successfully authenticated there are typically no further challenges to their identity, which leads to the question of how much confidence can we have that the authenticated user is who their credentials claim them to be? Behavioural biometrics *[2]* is an area that can be used to supplement traditional security methods with a continuous identification approach. Keystroke dynamics *[3]* captures a user's keystrokes as a means of confirming the identity of the current user. Traditional methods of identification (such as a password) rely on what the user has typed, in this instance whether the password entered matches the stored password. Keystroke dynamics instead focuses on how the user types, focusing on the timings of various key presses and releases. These timings can be used to build a unique identifier, a behavioural fingerprint of sorts, for an individual user.

This research extends the notion of keystroke dynamics beyond a simple process for confirming the identity of a specific user. Typically, keystroke dynamics is used to increase the confidence in the identity of an authenticated user. Instead, this work leverages a user's typing patterns to infer identity information such as name and native language.

This project aims to deliver an automated process for inferring identity cues (for example, name and native language) about an anonymous user of a computer service, based on the way that they type. It is hypothesised that an individual will type-specific combinations of characters, n-grams, more quickly than others depending on their familiarity with these groupings. This hypothesis is based on Fitts and Posner's *[4]* three-stage model of motor learning that suggests a skill becomes more automatic as familiarity with it increases. In this instance, this means that it is expected that a user's commonly typed combinations will be identifiable.

The research aims to answer three key questions:

1. Can we predict the name of an anonymous user based on an analysis of their typing patterns?
2. Is it possible to determine a user's native language based on their typing behaviours?
3. How long does it take for a set of bigrams to become visible in an individual's typing patterns?

The remainder of this report provides an overview of behavioural biometrics, with a particular focus on keystroke dynamics. This is followed by the methodology used for the experimental phase of the project. Finally, the report presents the findings alongside a discussion of the implication of the results.

RELATED WORK

Biometrics are unique and measurable characteristics that can be used to identify and describe an individual and will typically fall into two broad categories: physiological and behavioural *[5]*. Physiological traits are related to distinguishing characteristics of the body of an individual, for example, fingerprints, eyes (both iris and retinal images), and vein recognition. Conversely, behavioural biometrics relates to the innate traits and behaviours displayed by individuals. Examples of behavioural biometrics include keystroke dynamics (where a user's typing patterns are analysed), mouse dynamics (where a user's mouse movements are captured and analysed), and gait analysis.

Keystroke dynamics can be defined as the analysis of the way that an individual interacts with a keyboard, based on the timings of individual keystrokes *[3]*. The typing patterns that are displayed by an individual can be uniquely identifiable in the same way as a person's signature or handwriting *[6, 7]*.

The study of keystroke dynamics has been an active area of interest since the early 1990s [8] and is usually deployed in one of two ways. The first application analyses keystroke dynamics when an individual is typing fixed text. For example, this approach is typically used as a means of password hardening *[9]*. This approach to keystroke dynamics is typically looking to augment existing authentication methods. As well as the user providing something that they know (e.g. the password) the system will also analyse how they type, based on an enrolment period involving the password being typed multiple times. This approach makes use of keystroke dynamics to confirm the identity of the user based on the credentials that they have supplied as well as the innate traits that they have displayed.

The second application of keystroke dynamics is the analysis of free text. These applications are broadly divided into two categories: those that require specific software to be installed on a client machine *[10, 11]* and those that use a remote web-based method of data collection analysis. Messerman et al. *[12]* provide one such example of a remote, web-based collection and analysis approach to keystroke dynamics. This method of analysis is particularly attractive as it provides a low-cost, remote means of identification and analysis without the requirement for specialised hardware or specific software to be installed. Instead, data collection is accomplished by embedding code into any website that you have access to. This approach provides a means of continuous identification, where the user's identity is verified whenever they are actively typing. While it is expected that an individual will have identifiable typing behaviours, these behaviours are also liable to change. This could be due to increased familiarity with a keyboard or perhaps something more temporary, such as an injury or a change in mood *[13]*.

Typically, there are a limited number of features that can be extracted from keystroke data: dwell time, flight time, and the timings of various substrings of characters *[8]*. The dwell time is recorded with only a single keystroke and provides a measurement of the elapsed time between pressing and releasing a single key. The flight time measures the amount of time between releasing one key and pressing the next. The flight time requires the input of at least two keys and for experienced users, this is often a negative value, as there is normally an overlap between the original key being released and the next key being pressed. Finally, an n-gram refers to the time taken to a type combination of ‘n’ characters where the time is recorded between pressing the first key and releasing the ‘n-th’ key *[14]*. In addition to these timing values, other features such as mistake ratio and middle time can also be gathered for analysis *[15]*.

One of the most common uses of keystroke dynamics is to confirm the identity of an authenticated individual

[12]. This method is essentially a pattern-matching process where the current typing activities are compared to a stored user profile. This means that the user has their current typing behaviours analysed to determine whether they match the stored behaviours of the authenticated user. Essentially this process is confirming whether or not the authenticated user is who their credentials claim them to be.

Keystroke dynamics are not only used in user identification and have also been used to successfully identify a range of soft biometric traits, as introduced by Jain et al. *[16]*. Soft biometric traits can be thought of as characteristics that provide some information about the individual but that cannot be used to uniquely distinguish between two individuals. For example, keystroke dynamics has been used to determine a range of soft biometric traits including gender, handedness, or typing style *[17]*. Recent research has expanded this into other applications, such as detecting deception *[18]*. The research that we present in this paper builds on this idea of soft biometric traits by using keystroke dynamics to determine identity data about an anonymous individual.

This project evolves these nascent research areas to exploit keystroke dynamics to infer identity data about an unknown and anonymous user. This work is a fusion of the previously discussed methods, which could be used to not just confirm the identity of a user but also infer key identifying information about the user, such as name or native language. The aim of this research is a novel and innovative idea that looks to expand the existing areas of research in keystroke dynamics.

METHODOLOGY

EXPERIMENT 1: INFERRING IDENTITY CUES

INFERRING NAME

The core component of this research is the data collection framework used to capture the keystrokes and associated timings of the participants. A web-based, remote data-collection framework was developed for this research, which collected the following data:

- Forename
- Surname
- Age group
- Handedness
- Gender
- Native language

The participants were then required to copy a dynamically generated paragraph of text three times. The text is generated based on the participant's name. The forename and surname are segmented into two-character substrings (bigrams), for example:

- Participants name: John Smith
- Bigrams: jo, oh, hn, sm, mi, it, th

The generated bigrams are then used to create a paragraph of text that is specific to each participant, the code for generating the dynamic text based on a user's name is included in *Appendix A*. This approach ensures the data collection captures the participant typing the elements of their name, although not necessarily in the correct order. Without this approach, it is difficult to guarantee a user would type all of the elements of their name.

When completing the typing tasks, the website makes use of JavaScript keyboard listeners to ensure that whenever a key is pressed the software will capture:

- The key that was pressed
- The timestamp of when the key was pressed
- The timestamp of when the key was released

It is important to note that the users were prevented from pasting text into any of the boxes to ensure that keystroke patterns were collected.

The participants were largely recruited through social media, the researcher's professional and personal networks, and from the CREST community. Initially, data collection was planned to use a service such as Prolific or Amazon's Mechanical Turk. However, both services would not permit a participant's name to be collected and stored through their services owing to the associated privacy policies. This significantly impacted the number of participants that were recruited, with 84 participants being recruited, which is some way off the 500 that was initially planned.

DATA PREPARATION

Once the data had been collected a ranking of the 2-letter n-grams (bigrams) was created for each of the participants. This meant segmenting the data into bigrams and calculating the flight time, which is the time between releasing the first key and pressing the next.

The bigrams were ranked from the fastest to the slowest for each participant. Rankings are used rather than raw timings to normalise any variation between participants and their typing speed or experience. The research focuses on the relative speed between different bigrams rather than the absolute speed. This is based on the hypothesis that an individual user will

type particular n-grams faster than others based on their familiarity. For example, if a user has the n-gram 'iv' in their name then it is hypothesised that this n-gram will rank more highly than for a user without the n-gram in their name.

INFERRING NATIVE LANGUAGE

In addition to developing a model to predict the name of an anonymous user, this research aims to determine the native language of an individual based on their typing patterns. The data collection process is the same as that used when predicting names, where participants are required to provide demographic data and complete various typing tasks. During these tasks, the software will again capture the same metrics (key pressed, time pressed, and time released).

This experiment used Prolific as a tool for recruitment as it allows target recruitment of specific demographics, in this case, native language. Participants were recruited based on their native language, with five languages, which all use the Roman alphabet. The aim was to recruit an even distribution of participants across all five of the selected languages, with 100 participants in each group. Ultimately, we were able to collect 492 usable typing samples for this experiment owing to some data being corrupt or incomplete. The breakdown of participants collected is as follows:

- English – 92
- French – 100
- Spanish – 100
- Italian – 100
- German – 100

DATA PREPARATION

The data preparation is a similar process, with the typing data broken down into bigrams for each participant. Again, the bigrams are ranked based on their flight time, which is the time between when the first key is released and the second is pressed.

The hypothesis behind this process and preparation is similar in that each of the five languages has bigrams that appear more frequently than others. For example, 'th' is very common in the English language and as such participants that are used to using English will have a greater familiarity with this bigram. The experiment aims to leverage these common linguistic traits to predict an anonymous user's native language.

EXPERIMENT 2: UNDERSTANDING MOTOR-LEARNING

The second experiment aims to improve our understanding of the motor-learning phase of keystroke dynamics. The experiment uses a longitudinal approach to determine the impacts of repetition on a user's typing behaviours.

The study recruited 10 participants who were required to log into a website once per day for 10 days and complete a typing task each day. The daily tasks included two elements:

1. Participants were required to enter their username, which consisted of two randomly selected words, 10 times.
2. Copying a paragraph of text that contained the bigrams from the participant's username.

The first element is designed to encourage repetition to build familiarity with a particular set of bigrams. The second element then tests the typing behaviours of the individual each day to understand when repetition begins to become identifiable in these behaviours.

DATA PREPARATION

The experiment uses a similar method of data preparation and processing in that input data is broken down into bigrams. The bigrams are then ranked daily to highlight any notable changes in the rankings of the bigrams contained within the participant's username.

ANALYSIS AND RESULTS

EXPERIMENT 1: INFERRING IDENTITY CUES

INFERRING NAME

A stratified data resampling technique was adopted, with the data being resampled 30 times. During the resampling, the data is split into training and test splits at a ratio of 80% training data, which is used to build a model, and 20% test data, which is used to validate the model that has been produced.

While resampling the data is also standardised using only the training data, it should be noted that the data in this experiment is imbalanced with only approximately 5% of the labels in both the test and training dataset being labelled as true. This means that the number of bigrams that are in the typing data and that also appear in a user's name is very low. The data shows far more examples of bigrams that are not in the user's name, than examples of those that occur in a user's name.

A range of machine learning algorithms, and to ensure a fair comparison each algorithm was trained and evaluated on 30 resamples, where the resamples were the same across all algorithms. Hyperparameter optimisation is crucial to the performance of each algorithm and as such the hyperparameters were tuned for each of the algorithms that were used. A cross-validation grid search technique was used to choose the best parameter based on the training data alone. This ensured that the test data provided no influence on the selection of parameters.

Table 1 highlights the performance of the different algorithms that have been applied to predicting the bigrams in a user's name. It can be seen that some algorithms offer an excellent accuracy, however, owing

Classifier	In name (%)	False positive (%)	Accuracy (%)	Balanced acc. (%)
XGBoost (1)	45.30	8.33	88.74	**68.25**
XGBoost (2)	64.87	21.97	76.15	**70.83**
Decision Tree	27.91	4.17	91.97	**61.75**
K-NN	17.80	4.60	90.99	**56.46**
Naïve Bayes	12.18	8.98	86.31	**51.34**
SVM	53.57	19.78	77.73	**66.34**
AdaBoost	27.40	3.84	92.28	**61.67**
Random Forest	10.11	0.49	94.69	**54.80**

Table 1: A summary of results for different machine learning algorithms

to the sparsity of the training data this figure cannot be used in isolation when determining the best performing algorithm. For example, a decision tree appears to be a very accurate classifier with an accuracy of nearly 92%. However, in this instance, the algorithm was very good at predicting those bigrams that did not appear in the user's name.

A more representative measure of success is the algorithm's balanced accuracy. This provides a measure of accuracy in terms of bigrams that do appear in a name and those that do not. XGBoost produced the highest balanced accuracy and in name accuracy. The hyperparameters were crucial to this performance and the two XGBoost results highlighted in *Table 1* describe some interesting trends. Despite only being separated by a minimal margin in balanced accuracy, there is a notable difference in both accuracy and false positives predicted. This is important as in future work this underpinning approach could be used in a predictive system, and the sensitivity of false positives may be of particular importance.

Using the balanced accuracy, the XGBoost algorithm offers the best performance with 70% of an anonymous user's name based on the observation of their typing behaviours alone. *Appendix B* provides a detailed breakdown of the 30 resamples for this algorithm.

NATIVE LANGUAGE

As with the previous experiment to determine a participant's name, the time between when a key is released and the next key is pressed is recorded and assigned to the bigram value. There is a likelihood that these bigrams will be repeated, for example, the bigram 'en' will occur more than once in the typed passage of text. In this instance, where there are multiple instances of bigrams, the average (mean) time for every occurrence of that bigram is calculated.

When considering an individual's native language, the five most common bigrams for each language were used but owing to an overlap in bigram popularity across the five languages this resulted in a total of 15 bigrams. These bigrams are:

- th
- he
- in
- er
- an
- de
- es
- en
- el
- la
- le
- et
- il
- ch
- ei

Each participant completed a typing exercise, which captured the timings for each bigram, not just those listed above. A participant's bigrams are then ranked, again across all bigrams, with the rankings forming the input for the machine learning classification as these rankings help mitigate differences in typing speed and experience across the cohort of participants.

The data were resampled 30 times and used an 80:20 split, where 80% of the data was used in training and 20% used to evaluate the model. The data was standardised using the Scikit-learn StandScaler, which is used to normalise the range of feature variables.

Random under sampling was used in some experiments where appropriate. This randomly removes training data from the majority class, for example, where there are 50 positive cases and 75 negative cases, 25

negatives would be removed at random to ensure there are 50 in each class for training purposes. No sampling is applied to the test data and is still unbalanced as removing test data will not improve the results and may ultimately be considered unreliable.

Several machine learning classifiers have been used and for each of them, hyperparameter tuning has been performed. A cross-validated grid search was used to find the parameters that offer the best results on the training data alone. These parameters are then used to train and evaluate each of the classifiers. A classifier is evaluated using the test data, which it has not seen before the evaluation and that is not used during hyperparameter optimisation. The process is repeated for 30 different data resamples, that is to say, 30 different splits of training and test data. Each classifier uses the same data resamples to ensure a fair comparison between all classifiers.

ENGLISH VERSUS ALL RESULTS

The first test performed was to develop a model for distinguishing between English and the other four languages used. This model looks to determine whether an individual's native language is English or something else. Essentially, this will provide a binary decision between whether English is their native language or not.

The best average balanced accuracy was 71%, which was achieved using the SVC classifier, with a hyperparameter C value of 1000000 and gamma 1e-6. This provides a significantly better result than a random guess, and there is scope for further improving this result with a greater volume of data and an improved training model.

NATIVE LANGUAGE PREDICTION

The next stage was to develop a model to classify a user's native language as English, French, German, Spanish, or Italian based on their typing behaviours.

The best average balanced accuracy was 45%, which was again achieved using the SVC classifier. The hyperparameter C value was 10 and a gamma value of 0.01.

While the accuracy is not as good as the simple binary decision of English or another, 45% represents a better performance than a random guess.

EXPERIMENT 2: UNDERSTANDING MOTOR LEARNING

The experiment provides an initial investigation into understanding the process of motor learning regarding typing behaviours. This first iteration required a small group of participants to complete a longitudinal study, where they repeated a particular phrase every day for 10 days. During this time the keystroke timings were recorded, as with the previous experiments; the experiment records the difference between releasing the first key and pressing the second.

In line with previous experiments, the bigrams typed by an individual were ranked based on the time it took to type them. Again, this was to normalise any individual differences in typing speed or experience across the range of participants. After the completion of each task, a ranking of bigrams was created for each of the participants, with the change in rankings used as a metric for a change (increased or decreased) in familiarity.

Appendix D shows a graph for each of the individual participants and the change in rankings for the bigrams in their unique phrase, over the 10 days of the experiment. The most interesting thing to note is that 7 of the 10 participants had a notable change in ranking for the bigrams contained in their assigned phrase. This means that, on average, the bigrams contained in their assigned phrase became quicker as they progressed through the tasks.

CONCLUSIONS AND FUTURE WORK

CONCLUSIONS AND DISCUSSION

This work has identified a novel application of keystroke dynamics. Whereas previous work has focused mainly on identifying an individual based on the way that they type, this work extends these ideas by predicting identity cues, such as name or native language.

The approach to predicting the bigrams in the name of an individual has proved successful, with a balanced accuracy of 71%. The biggest area for improvement for this result would be to collect a larger volume of data when building the machine learning model in the first instance. The initial aim was to collect typing data from around 500 individuals, the reality was that restriction in services (e.g. Prolific and Amazon's Mechanical Turk) meant that it was difficult even to get close to this volume of data. The final count of 84 participants was far lower than expected but the results show that this is indeed a promising approach.

When considering the native language of an anonymous user the experiment was able to work with a much larger cohort of 492 participants in total. The most promising result was when considering native language as a binary choice, that is to say between English or another language. The balanced accuracy of 70% shows a good level of accuracy when determining whether English is the user's native language.

However, when trying to predict the difference between one of five languages (English, French, German, Spanish, Italian) the accuracy is greatly reduced (to approximately 45%). While this result is still better than a random guess, it is less than ideal. The inaccuracy could be due to various factors, for example, several of the languages chosen share common bigrams. Changes could also be made to the way that the data is modelled when considering which popular bigrams to include.

The second experiment, which focused on understanding the motor-learning process for typing patterns, yielded some promising results. This was only an initial pilot with a limited number of participants, but it showed that for the majority of participants there was a discernible difference in their typing behaviours within the space of 10 days.

FUTURE WORK

Several elements can be built upon going forwards to further improve the results achieved here, or to drive this work in different directions.

- Perhaps the most obvious area for future work is to look to build a larger, more diverse, cohort of data to produce a completer and more accurate machine learning model. Based on the experiences with this project, a larger volume of training data would certainly improve the accuracy of our predictions going forwards.
- Predicting whether a user was a native English speaker or not delivered good results (70%), however, this did not carry over when trying to predict the language as a choice of five possible classes. There are optimisations or other avenues to explore for data processing and collection.
- The focus in these experiments was on languages that use the Roman alphabet. It would provide an interesting challenge to include other languages, which are not based on this alphabet.
- At present we have focused on bigrams, but there may be merit in exploring variable-length n-grams to try and find the optimal point for identification.

- Currently, the work aims to extract identity cues that are fairly commonplace and benign. Exploring further cues, such as passwords, would provide an interesting challenge.
- The research carried out to date focuses on physical keyboards. As technology and our use of it continues to evolve, there is an increased reliance on smartphones and tablets. The majority of these personal devices use a virtual keyboard on a touchscreen and so the logical evolution of this work would be to attempt to apply the same principles to touchscreens. This could also leverage new paradigms for typing, such as swipe keyboards.
- Typing is a single factor in a whole range of behavioural biometrics. An interesting next step for this work would be to consider a range of other factors. This could include a mouse, touchpad, interaction with a user interface, or swiping.

REFERENCES

[1] E. F. Gehringer, Choosing passwords: Security and Human Factors in Technology and Society, 2002 (ISTAS'02). 2002 International Symposium on. IEEE, 2002, pp. 369–37

[2] R. V. Yampolskiy and V. Govindaraju, Behavioural biometrics: a survey and classification, International Journal of Biometrics, vol. 1, no. 1, pp.81–113, 2008

[3] F. Monrose and A. D. Rubin, Keystroke dynamics as a biometric for authentication, Future Generation computer systems, vol. 16, no. 4, pp.351–359, 2000

[4] P. M. Fitts and M. I. Posner. Human performance. 1967

[5] K. Delac and M. Grgic, A survey of biometric recognition methods in Electronics in Marine, 2004. Proceedings Elmar 2004. 46th International Symposium. IEEE, 2004, pp. 184–193

[6] S. Douhou and J. R. Magnus, The reliability of user authentication through keystroke dynamics, Statistica Neerlandica, vol. 63, no. 4, pp.432–449, 2009

[7] A. Dvorak, N. L. Merrick, W. L. Dealey, and G. C. Ford, Type writing behaviour, New York: American Book Company, vol. 1, no. 6, 1936

[8] S. Bleha, C. Slivinsky, and B. Hussien, Computer-access security systems using keystroke dynamics, IEEE Transactions on pattern analysis and machine intelligence, vol. 12, no. 12, pp. 1217–1222, 1990

[9] D. Rudrapal, S. Das, and S. Debbarma, Improvisation of biometrics authentication and identification through keystrokes pattern analysis, in International Conference on Distributed Computing and Internet Technology. Springer, 2014, pp. 287–292

[10] M. Rybnik, M. Tabedzki, and K. Saeed, A keystroke dynamics-based system for user identification, in Computer Information Systems and Industrial Management Applications, 2008. CISIM'08. 7th. IEEE, 2008, pp. 225–230

[11] P. Pinto, B. Patrao, and H. Santos, Free-typed text using keystroke dynamics for continuous authentication, in IFIP International Conference on Communications and Multimedia Security. Springer, 2014, pp.33–45

[12] A. Messerman, T. Mustafic, S. A. Camtepe, and S. Albayrak, Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics, in Biometrics (IJCB), 2011 International Joint Conference on. IEEE, 2011, pp. 1–8

[13] C. Epp, M. Lippold, and R. L. Mandryk, Identifying emotional states using keystroke dynamics, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2011, pp. 715–724

[14] F. Bergadano, D. Gunetti, and C. Picardi, User authentication through keystroke dynamics, ACM Transactions on Information and System Security (TISSEC), vol. 5, no. 4, pp. 367–397, 2002

[15] R. Giot, M. El-Abed, and C. Rosenberger, Keystroke dynamics overview, in Biometrics. In Tech, 2011, pp. 157–182

[16] A. K. Jain, S. C. Dass, and K. Nandakumar, Soft biometric traits for personal recognition systems, in Biometric Authentication. Springer, 2004, pp. 731–738

[17] S. Z. S. Idrus, E. Cherrier, C. Rosenberger, and P. Bours, Soft biometrics for keystroke dynamics:

Profiling individuals while typing passwords, Computers & Security, vol. 45, pp. 147–155, 2014

[18] M. Monaro, R. Spolaor, Q. Li, M. Conti, L. Gamberini, and G. Sartori, Type me the truth! Detecting deceitful users via keystroke dynamics, in Proceedings of the 12th International Conference on Availability, Reliability and Security, 2017, pp. 1–6

APPENDIX A

AutoClicka.py

```
import pickle
import logging
import re
import itertools
class AutoClicka:
 def __init__(self, sentences_file="sentences.
pickle"):
 logging.debug("Building AutoClicka object")
 logging.debug("Loaded pickle file")
 self.sentences = pickle.load(open(sentences_
file, 'rb'))
 self.correct_format = re.compile('[a-z][a-z]')
 def generate_sentences(self, name, level=1):
 name = name.lower()
 logging.debug("Generating sentence for %s" %
name)
 # Pull out the bigrams in the name
 bigrams_in_name = []
 for ii in range(0, (len(name) - 1)):
 bigram = name[ii:(ii + 2)]
 if self.correct_format.match(bigram):
 bigrams_in_name.append(bigram)
 bigrams_in_name = [
 'ik', 'ur', 'it', 'id', 'lu', 'ev', 'nd', 'wa', 'fa', 've', 'rl', 'to',
'ke', 'ry', 'ea',
 'so', 'yn', 'ba', 'jo', 'ed', 'et', 'us', 'st', 'yl', 'te', 'ch',
'ei', 'ce', 'ca', 'th',
 'im', 'av', 'ac', 'em', 'va', 'ti', 'ab', 'ly', 'za', 'he', 'be',
'at', 'di', 'ol', 'si',
 'iy', 'vi', 'ir', 'hi', 'lo', 'de', 'es', 'ey', 'ae', 'or', 'me',
'ta', 'ad', 'nn', 'da',
 'os', 'aa', 'as', 'ne', 'ka', 're', 'mi', 'ja', 'll', 'ni', 'se',
'il', 'sh', 'on', 'sa',
 'ee', 'ro', 'er', 'is', 'ai', 'am', 'in', 'en', 'al', 'ay', 'ia',
'ri', 'ya', 'ie', 'ah',
 'le', 'el', 'li', 'la', 'na', 'ha', 'ra', 'ma', 'ar', 'an']
 bigram_pairs = set()
 # Find bigram pairs in list
 for ii in range(0, (len(bigrams_in_name) - 1), 2):
 bigram_pairs.add(''.join(sorted(bigrams_in_
name[ii:(ii + 2)])))
 # Need to catch if we have a hanging bigram we
miss 50 % of the time
 # this will not be required and add a duplicate to
the set but it's a
 # set so who gives a toss
 bigram_pairs.add(''.join(sorted(bigrams_in_
name[-2:])))
 print("pairs:", bigram_pairs)
 content = set()
 for bigram_pair in bigram_pairs:
 if bigram_pair in self.sentences:
 content.add(self.sentences[bigram_pair])
 else:
 logging.warning("May be missing bigram for %s"
% bigram_pair)
 # Finding minimum length of content which
covers all the bigrams
 # TODO probably want to tidy this !
 print(content)
 print(len(content) + 1)
 counts = [0] * len(bigrams_in_name)
 putative_story = []
 found_all = False
 content = ['the arrival of hubert and emily saved
the small boy from many a cuff and the '
 'donkey from a kick or two; and jack stood amid
the ruin he had created, as '
 'quiet and as docile a creature as the mind could
imagine.',
 'he did not dare intimate his change of mind to
his sister; but the news '
 'having reached mrs. price in various rumours,
she wrote to her brother '
 'asking him to confirm or deny these rumours;
and when he admitted their '
```

```
 'truth, mrs. price never spoke to him again.',
 'he plunged into calculation of the time it would take him to finish it if he '
 'were to sit at home all day, working from seven to ten hours every day.',
 'in such reverie and such consideration he lay immersed, oblivious of the '
 'present moment, and did not stir from his chair until the postman shook the '
 'frail walls with a violent double knock.',
 'there were piles of newspapers, there were books on the mahogany sideboard '
 'and on the horsehair sofa, and on the table there were various '
 'manuscripts,--_the gipsy_, act i.; _the gipsy_, act iii., scenes iii.',
 'i can imagine these women living in admiration of this man, tending on him, '
 'speaking very little, removed from worldly influences, seeing only the young '
 "men who come every tuesday evening to listen to the poet's conversation--i "
 "don't hear them saying much--i can see them sitting in a corner listening "
 'for the ten thousandth time to aestheticisms not one word of which they '
 "understand, and about ten o'clock stealing away to some mysterious chamber.",
 'the indians then assaulted them with a hideous noise: "sixty or seventy of '
 'them, some black, some red, some white, some particoloured, came in a square '
 'order, singing and dancing out of the woods, with their okee (which is an '
 'idol made of skinnes, stuffed with mosse, and painted and hung with chains '
 'and copper) borne before them; and in this manner being well armed with '
 'clubs, targets, bowes and arrowes, they charged the english that so kindly '
 'received them with their muskets loaden with pistol shot, that down fell '
 'their god, and divers lay sprawling on the ground; the rest fled againe to '
 'the woods, and ere long sent men of their quiyoughkasoucks [conjurors] to '
 'offer peace and redeeme the okee."',
 'as we pass around the house, i discover a boy in the ravine filling a bag '
 'with chestnuts and hickorynuts.',
 'did those who believed in the old formulas imagine that the new formula '
 'would be discovered straight away, without failures preliminary?',
 'emily stopped before a bed-room, and, looking at hubert shyly and '
 "interrogatively, she said-- 'this is my room.",
 'a zig-zag fugitive thought passed: why did the fly-man speak of taking them '
 'to the station?',
 'but at the end of a week his health began to give way, and, like a man after '
 'a violent debauch, he thought of returning to a more normal existence.',
 'had the excellent rip van winkle, instead of seeking his repose upon the '
 'cold and barren acclivities of the kaatskills--as we are veritably informed '
 "by irving--but betaken himself to a comfortable bed at morrison's or the "
 'bilton, not only would he have enjoyed a more agreeable siesta, but, what '
 'the event showed of more consequence, the pleasing satisfaction of not being '
 'disconcerted by novelty on his awakening.',
 'she returned soon after with a small basket; and a large retriever, tied up '
 'in the corner of the yard, barked and lugged at his chain.',
 'and a few days after she sent annie with a note, reminding him of his '
 'promise to read her what he had written.']
 story = []
 # for ii in range(16,len(content)+1):
 # if not found_all:
```

```
 # logging.debug("Reducing - trying level %d" %
ii)
 # for putative_story in itertools.
permutations(content, ii):
 counts = [0] * len(bigrams_in_name)
 for ii in range(level, len(content) + 1):
 logging.debug("Reducing - trying level %d" % ii)
 all_found = 0
 for putative_story in itertools.
permutations(content, ii):
 all_found = 0
 counts = [0] * len(bigrams_in_name)
 for sentence in putative_story:
 useful = False
 all_found = 0
 for i, bi in enumerate(bigrams_in_name):
 if counts[i] == 0:
 count = sentence.count(bi)
 if count > 0:
 useful = True
 counts[i] = count
 else:
 all_found += 1
 if useful:
 story.append(sentence)
 if all_found == len(bigrams_in_name):
 break
 if all_found == len(bigrams_in_name):
 print("FOUND")
 break
 if all_found == len(bigrams_in_name):
 print("FOUND")
 break
 # print(bi)
 # count = sentence.count(bi)
 # if count > 0:
 # counts[i] += count
 # putative_story.append(sentence)
 # found_all = True
 # for count in counts:
 # if count == 0:
 # found_all = False
 print(len(putative_story))
 print(counts)
 return list(putative_story)
```

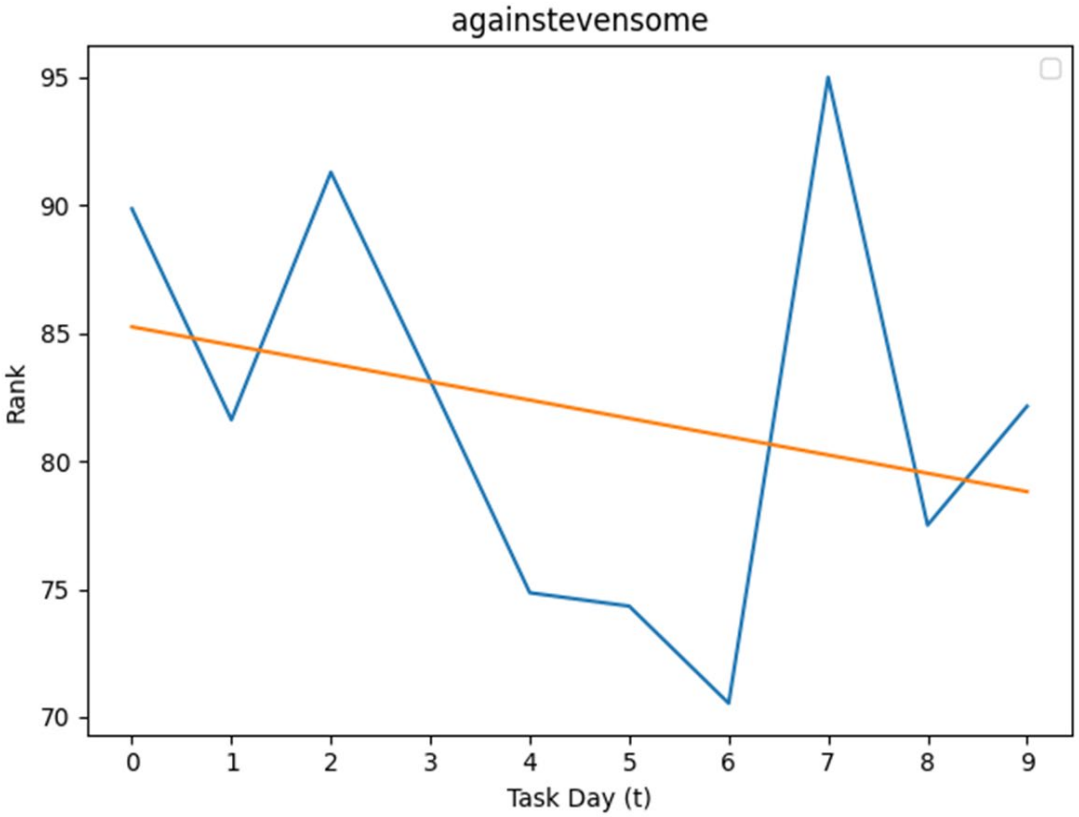
APPENDIX B

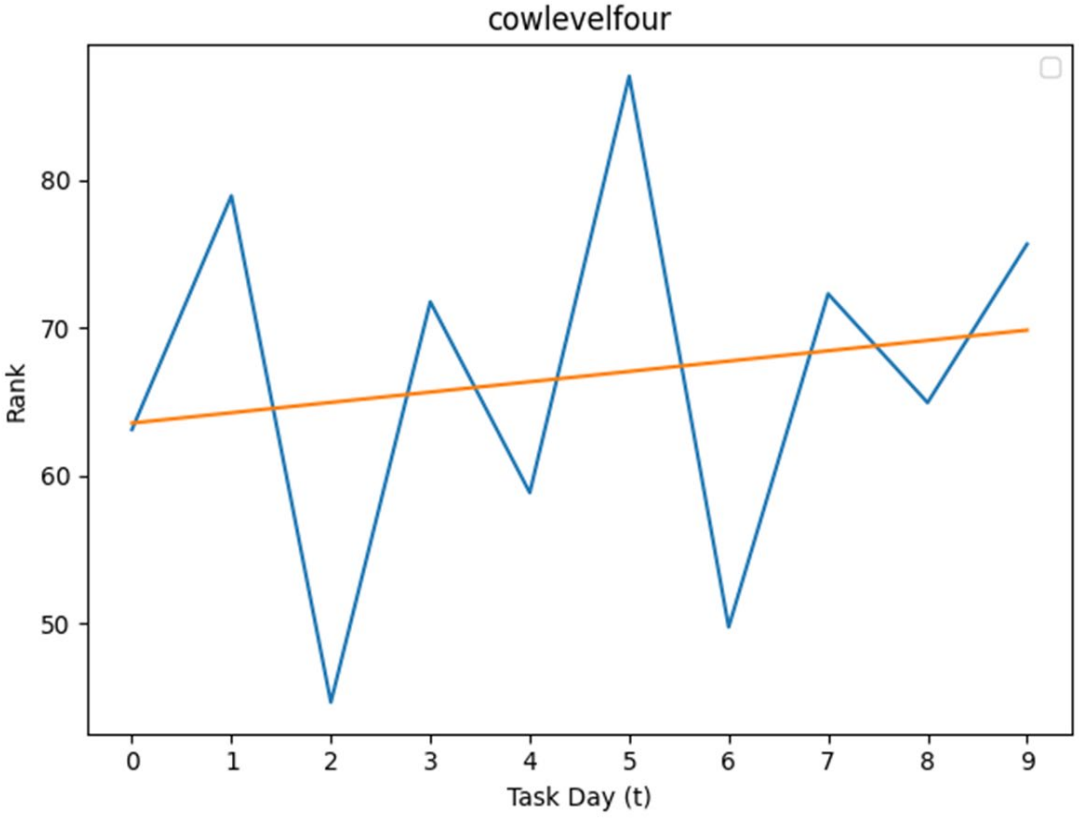
Test	corr	fps	acc	Balacc	f1_av	prec_0	prec_1	rec_0	rec_1	f1_0	f1_1
1	0.6420	0.2143	0.7661	0.7076	0.5466	0.9739	0.1410	0.7732	0.6420	0.8620	0.2312
2	0.6599	0.2223	0.7592	0.7124	0.5435	0.9750	0.1391	0.7649	0.6599	0.8573	0.2298
3	0.6580	0.2117	0.7694	0.7169	0.5520	0.9748	0.1467	0.7759	0.6580	0.8641	0.2400
4	0.6438	0.2253	0.7568	0.7033	0.5332	0.9759	0.1256	0.7628	0.6438	0.8563	0.2101
5	0.6542	0.2247	0.7568	0.7084	0.5398	0.9750	0.1349	0.7626	0.6542	0.8558	0.2237
6	0.6177	0.2198	0.7591	0.6925	0.5392	0.9717	0.1345	0.7674	0.6177	0.8575	0.2208
7	0.6641	0.2187	0.7632	0.7165	0.5460	0.9757	0.1406	0.7688	0.6641	0.8600	0.2321
8	0.6772	0.2211	0.7620	0.7219	0.5445	0.9772	0.1384	0.7667	0.6772	0.8592	0.2298
9	0.6247	0.2145	0.7646	0.6988	0.5447	0.9721	0.1397	0.7729	0.6247	0.8611	0.2283
10	0.6071	0.2218	0.7577	0.6866	0.5319	0.9726	0.1247	0.7660	0.6071	0.8570	0.2068
11	0.6662	0.2095	0.7725	0.7224	0.5530	0.9762	0.1461	0.7786	0.6662	0.8663	0.2397
12	0.6432	0.2248	0.7553	0.7026	0.5407	0.9731	0.1376	0.7619	0.6432	0.8547	0.2267
13	0.6414	0.2240	0.7561	0.7022	0.5406	0.9731	0.1371	0.7629	0.6414	0.8553	0.2259
14	0.6710	0.2334	0.7493	0.7123	0.5353	0.9763	0.1316	0.7536	0.6710	0.8506	0.2201
15	0.6592	0.2214	0.7609	0.7128	0.5406	0.9762	0.1338	0.7665	0.6592	0.8587	0.2225
16	0.6394	0.2138	0.7665	0.7066	0.5466	0.9737	0.1408	0.7738	0.6394	0.8623	0.2308
17	0.6457	0.2220	0.7598	0.7059	0.5375	0.9755	0.1303	0.7660	0.6457	0.8581	0.2168
18	0.6675	0.2316	0.7510	0.7116	0.5356	0.9763	0.1312	0.7556	0.6675	0.8519	0.2192
19	0.6398	0.2127	0.7685	0.7077	0.5439	0.9751	0.1357	0.7756	0.6398	0.8640	0.2239
20	0.6789	0.2178	0.7648	0.7243	0.5497	0.9766	0.1446	0.7697	0.6789	0.8609	0.2385
21	0.6655	0.2154	0.7645	0.7182	0.5566	0.9731	0.1561	0.7708	0.6655	0.8602	0.2529
22	0.6501	0.2231	0.7589	0.7075	0.5372	0.9758	0.1301	0.7648	0.6501	0.8575	0.2169
23	0.6584	0.2275	0.7545	0.7091	0.5372	0.9757	0.1321	0.7598	0.6584	0.8543	0.2200
24	0.6330	0.2129	0.7678	0.7042	0.5431	0.9744	0.1351	0.7753	0.6330	0.8635	0.2226
25	0.6519	0.2167	0.7650	0.7116	0.5440	0.9755	0.1371	0.7713	0.6519	0.8614	0.2265
26	0.6468	0.2216	0.7597	0.7064	0.5399	0.9748	0.1340	0.7660	0.6468	0.8579	0.2220
27	0.6459	0.2172	0.7636	0.7081	0.5445	0.9743	0.1389	0.7703	0.6459	0.8604	0.2286
28	0.6156	0.2146	0.7654	0.6946	0.5384	0.9734	0.1300	0.7736	0.6156	0.8621	0.2147
29	0.6939	0.2196	0.7642	0.7310	0.5493	0.9782	0.1436	0.7681	0.6939	0.8605	0.2380
30	0.5997	0.2165	0.7621	0.6855	0.5362	0.9715	0.1291	0.7713	0.5997	0.8599	0.2124

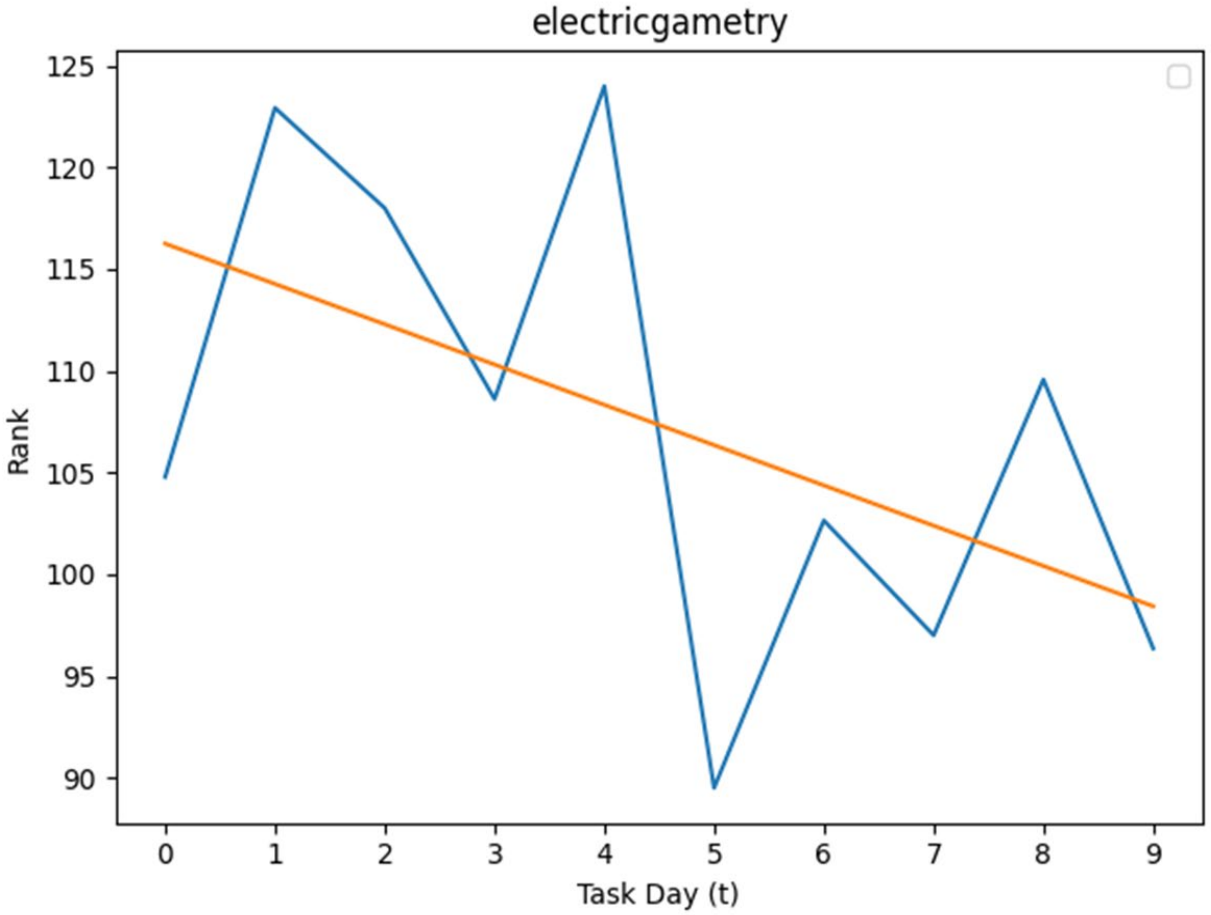
APPENDIX C

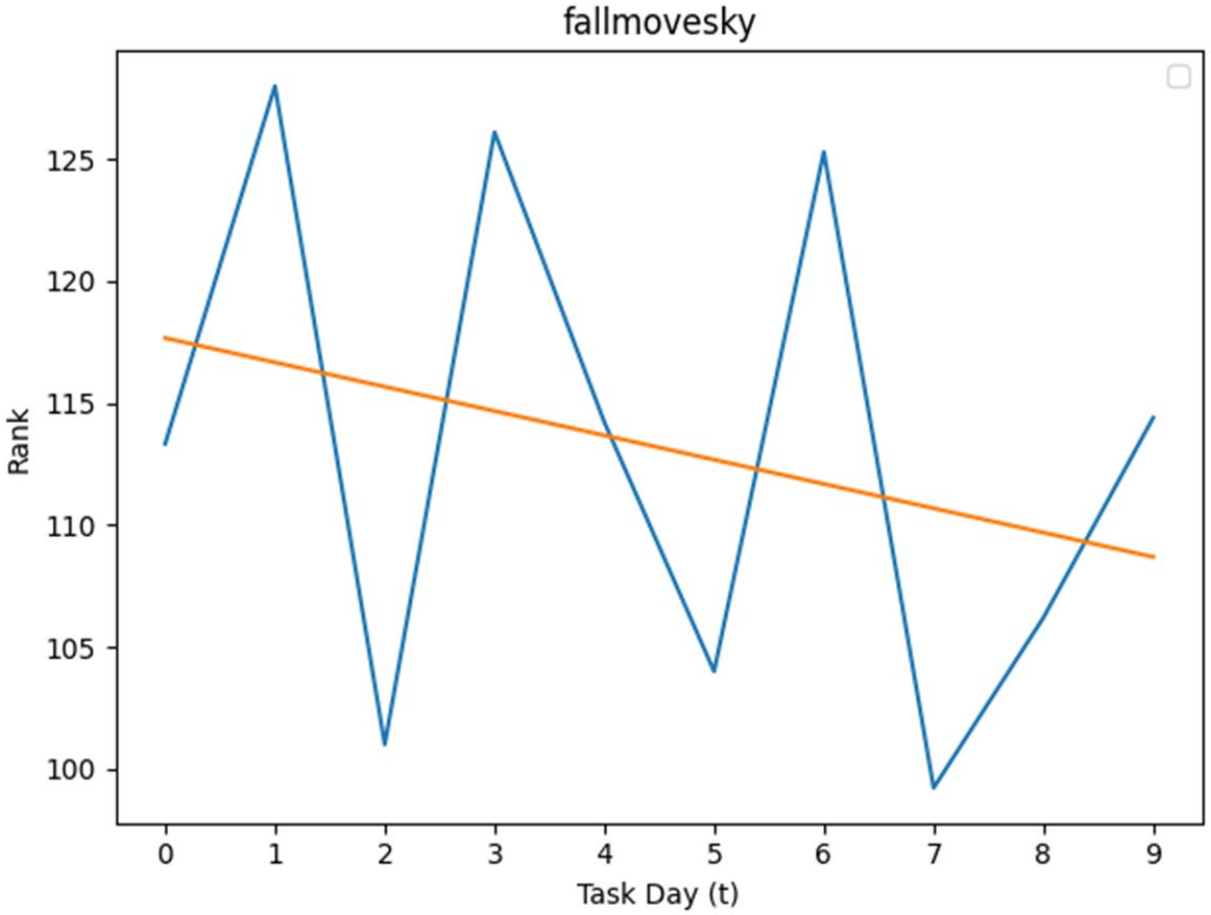
Resample	Accuracy	Balanced Accuracy	F1	F1 Macro	Precision	Recall
1	0.6765	0.6852	0.6765	0.6435	0.6765	0.6765
2	0.7255	0.7303	0.7255	0.6756	0.7255	0.7255
3	0.7745	0.7660	0.7745	0.7258	0.7745	0.7745
4	0.7745	0.7966	0.7745	0.7403	0.7745	0.7745
5	0.7843	0.7636	0.7843	0.7230	0.7843	0.7843
6	0.7255	0.7426	0.7255	0.6756	0.7255	0.7255
7	0.7647	0.6852	0.7647	0.6731	0.7647	0.7647
8	0.6765	0.6618	0.6765	0.6521	0.6765	0.6765
9	0.6471	0.6026	0.6471	0.5750	0.6471	0.6471
10	0.7255	0.7304	0.7255	0.6912	0.7255	0.7255
11	0.7843	0.7529	0.7843	0.7230	0.7843	0.7843
12	0.7353	0.7444	0.7353	0.7120	0.7353	0.7353
13	0.7255	0.7563	0.7255	0.7064	0.7255	0.7255
14	0.7059	0.6974	0.7059	0.6691	0.7059	0.7059
15	0.7255	0.7412	0.7255	0.6474	0.7255	0.7255
16	0.6373	0.6619	0.6373	0.6003	0.6373	0.6373
17	0.7059	0.6914	0.7059	0.6386	0.7059	0.7059
18	0.7059	0.7431	0.7059	0.6985	0.7059	0.7059
19	0.6961	0.6651	0.6961	0.6304	0.6961	0.6961
20	0.7157	0.6554	0.7157	0.6390	0.7157	0.7157
21	0.7451	0.7417	0.7451	0.7173	0.7451	0.7451
22	0.6863	0.7011	0.6863	0.6357	0.6863	0.6863
23	0.6373	0.6807	0.6373	0.6141	0.6373	0.6373
24	0.7255	0.7242	0.7255	0.6474	0.7255	0.7255
25	0.6176	0.5700	0.6176	0.5597	0.6176	0.6176
26	0.7059	0.7333	0.7059	0.6886	0.7059	0.7059
27	0.7255	0.7303	0.7255	0.6756	0.7255	0.7255
28	0.7549	0.7352	0.7549	0.7178	0.7549	0.7549
29	0.7353	0.7404	0.7353	0.6900	0.7353	0.7353
30	0.7059	0.6906	0.7059	0.6691	0.7059	0.7059

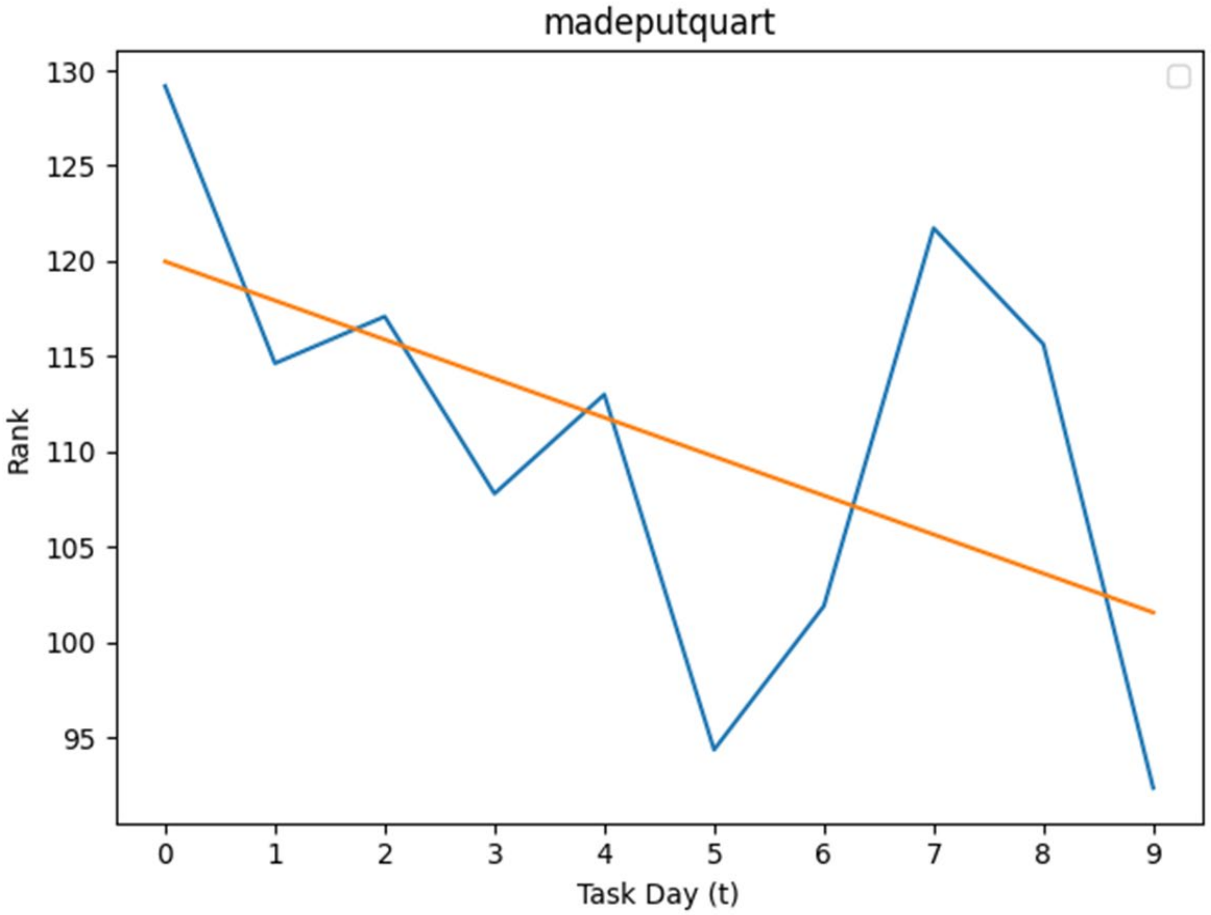
APPENDIX D

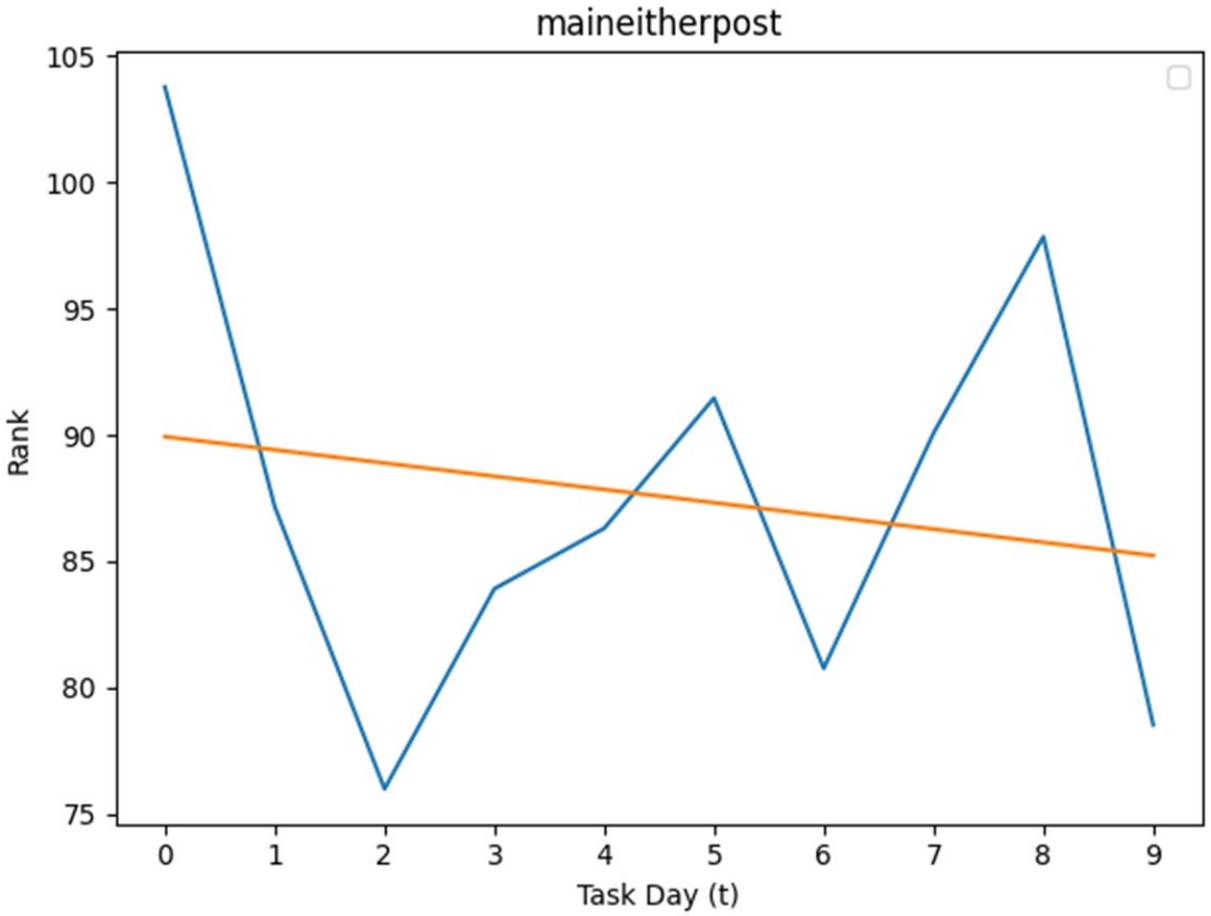

againstevensome
Rank
Task Day (t)
95
90
85
80
75
70
0
1
2
3
4
5
6
7
8
9

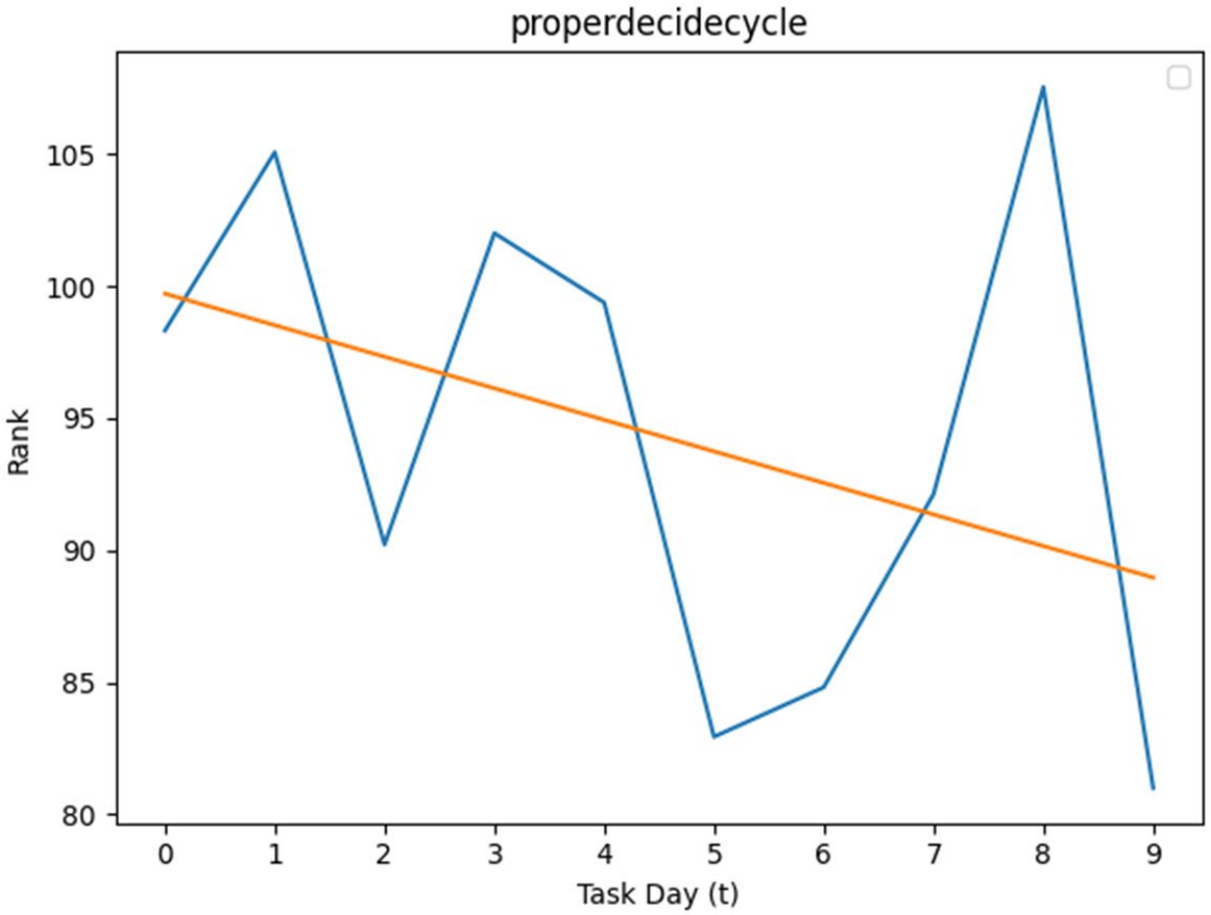

cowlevelfour
Rank
Task Day (t)
80
70
60
50
0
1
2
3
4
5
6
7
8
9

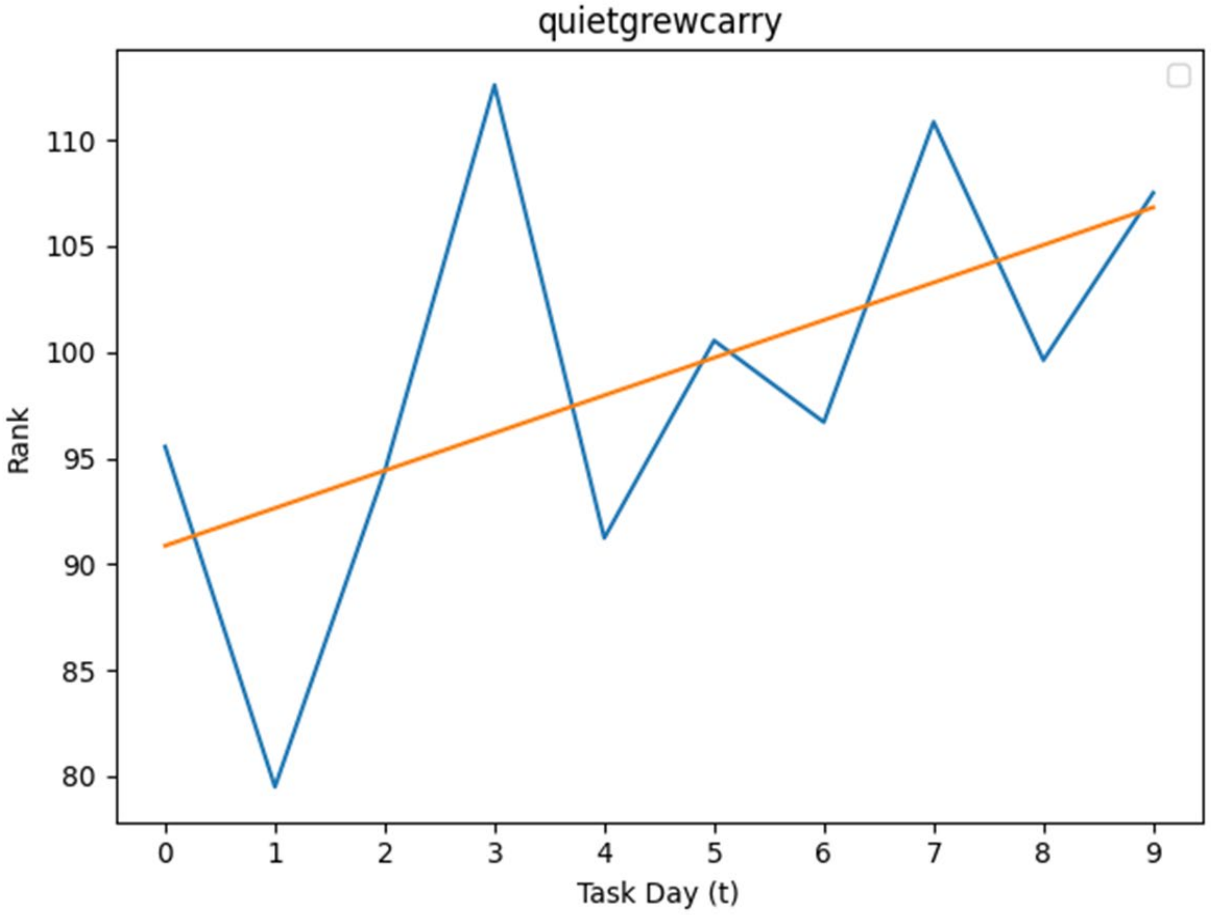

electricgametry
Rank
Task Day (t)
125
120
115
110
105
100
95
90
0
1
2
3
4
5
6
7
8
9

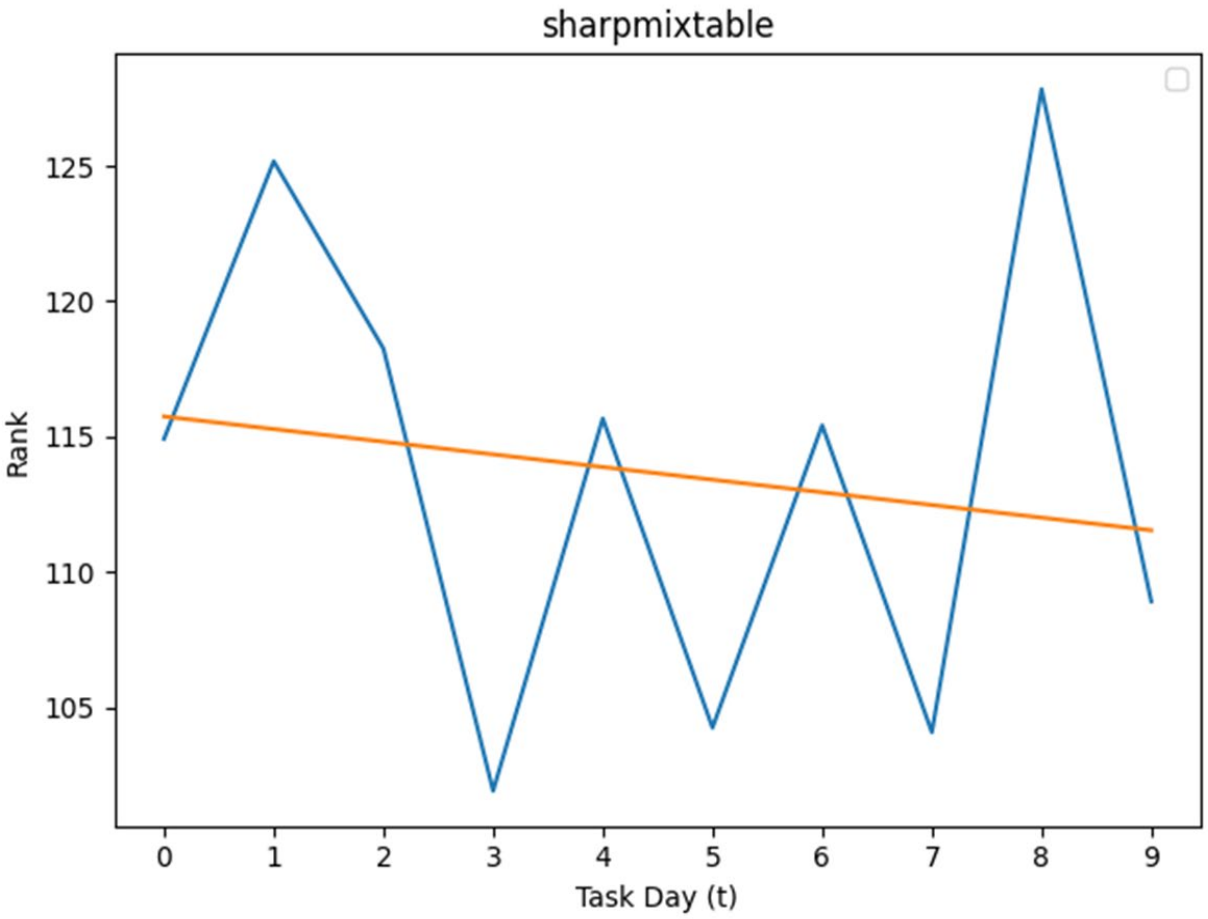

fallmovesky
Rank
Task Day (t)
125
120
115
110
105
100
0
1
2
3
4
5
6
7
8
9

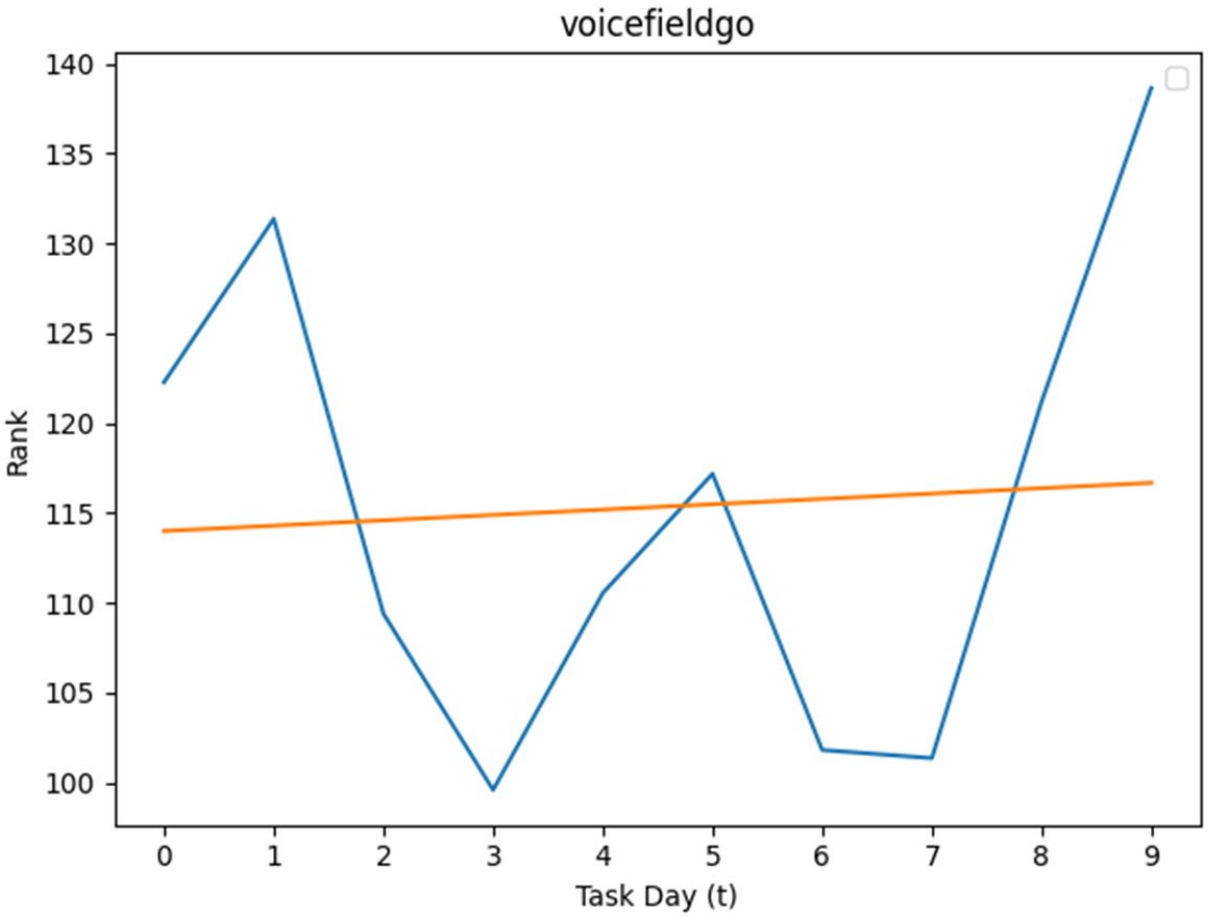

madeputquart
Rank
Task Day (t)
130
125
120
115
110
105
100
95
0
1
2
3
4
5
6
7
8
9


maineitherpost
Rank
Task Day (t)
105
100
95
90
85
80
75
0
1
2
3
4
5
6
7
8
9


properdecidecycle
Rank
Task Day (t)
105
100
95
90
85
80
0
1
2
3
4
5
6
7
8
9


quietgrewcarry
Rank
Task Day (t)
110
105
100
95
90
85
80
0
1
2
3
4
5
6
7
8
9


sharpmixtable
Rank
125
120
115
110
105
0
1
2
3
4
5
6
7
8
9
Task Day (t)


voicefieldgo
Rank
140
135
130
125
120
115
110
105
100
0
1
2
3
4
5
6
7
8
9
Task Day (t)

For more information on CREST
and other CREST resources, visit
www.crestresearch.ac.uk

CENTRE FOR RESEARCH AND
EVIDENCE ON SECURITY THREATS

21-036-01