12-2-2021 1:30 PM

Smart Chatbot For User Authentication

Peter Voege, *The University of Western Ontario*

Supervisor: Ouda, Abdelkader, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

Recommended Citation

Voege, Peter, "Smart Chatbot For User Authentication" (2021). *Electronic Thesis and Dissertation Repository*. 8315.
https://ir.lib.uwo.ca/etd/8315

Abstract

The field of authentication has a lot of room to develop in the age of big data and machine learning. Conventional high-accessibility authentication mechanisms including passwords or security questions struggle with critical vulnerabilities, creating a need for alternative authentication mechanisms able to cover said weaknesses.

We sought to create an authentication mechanism that creates dynamic, ever-changing security questions only the user can answer while remaining intuitive to use and as accessible as typical security questions by creating an authentication chatbot that leverages big data and natural language processing to pose dynamic authentication challenges.

We tested the components of our design in simulated scenarios to prove their efficacy, and found that all critical elements of the design can satisfactorily complete the tasks set for them. Thus we believe this design offers a useful supplement or alternative to password or security question-based authentication, improving the security of user data in our society.

Lay Summary

Passwords are vulnerable and insecure, and it's hard to fix that. Easily guessed passwords, password reuse, leaked databases, they're simple to use but carry not-insignificant risk.

Many modern systems improve security by layering multiple authentication mechanisms together, adding security questions or codes delivered via text message. Here we invent a new mechanism for this purpose, a chatbot designed to be convenient and straightforward to use yet secure and reliable.

The chatbot functions by taking unusual transactions or other recorded events about a user, ones that they're likely to remember, and automatically forming questions that only the real user can correctly answer. Artificial intelligence is extensively used in the process, allowing the system to get better and better the more people it authenticates.

Acknowledgements

I would like to express my thanks to the following people:

My family, for raising me into the person I am today and supporting me all the while. Particular thanks for their care and support in this time of global pandemic.

My colleagues Iman and Wafaa for their assistance and support during my graduate program.

My supervisor Dr. Ouda for everything he has done to teach and guide me throughout the last few years.

To all of you I give my sincere gratitude and wish you the best in your own endeavours.

Keywords: Authentication, Chatbots, Natural Language Processing

Contents

List of Figures

Chapter 1

Introduction

Digital authentication is the process of determining the true identity of a user attempting to access an electronic system, typically to ensure that secure information is only accessed by the people who ought to be accessing it. While determining someone's identity in real life may be easy due to how many personal characteristics (such as facial structure, voice, or body type) are highly persistent and difficult to fake, it is much less trivial to determine someone's identity online. The source of the online request can only tell you which device the request was sent from, which does not necessarily imply that the correct person is operating it, and as such more information must be gathered from the person operating the device.

Authentication mechanisms can be broken down into four main categories: ’something you know’, ’something you have’, ’something you are’, and ’something you do’. ’Something you know’ authentication queries the user for specific knowledge, for which a correct answer is evidence of authenticity and an incorrect answer is evidence of a fraudulent user. ’Something you have’ authentication uses physical objects that are expected to only be in the hands of authentic users in order to determine authenticity. If the user can show that they have that item, they are treated as authentic users. A good example of 'something you have' authentication is keys, such as for one's car or house. 'Something you are' authentication relies on immutable physical characteristics of the user in order to determine authenticity. This includes 'Something you do' authentication examines user behavioural patterns, such as typing or voice rhythm, in order to determine authenticity, under the principle that it is difficult to properly replicate things like another person's typing style.

1.1 Research Motivation

There are multiple distinct qualities that a good authentication system should have, and different authentication mechanisms perform differently for each quality[16]. The first and most central quality to authentication is security, or how well the system performs at its main task of discerning between authentic and fraudulent users. The second main quality that a good authentication system should have is accessibility, as an authentication system that requires things that the platform or user cannot provide will be unusable to that platform or user. Custom authentication hardware such as retina scanners is a good example of authentication mechanisms which are relatively inaccessible. The final key quality for authentication mechanisms to optimize for is convenience, as it is preferable to spend as little time and effort on authentication as possible, which results in a better user experience.

Many authentication systems make trade-offs between those three qualities, creating mechanisms that are convenient and accessible but insecure or that are convenient and secure but inaccessible. For example, 'something you do' authentication tends to be highly secure as it can be very hard to fake someone's physical characteristics or precise behavioural patterns, but applications which can authenticate based on these qualities tend to need specialized hardware that limit where and when it can be used[2][7]. As such, the optimal authentication mechanism is often context-sensitive, depending on the relative importance of each characteristic. Many systems found in day to day life do not hold exacting standards for security, and as a result lean towards solutions which are more accessible and convenient in order to make sure that as many people as possible may use the system and have a good user experience from it.

The most common authentication system that is both accessible and convenient is password-based authentication, which is not very secure. Not only are some passwords easily deduced or brute-forced, but the way passwords persist for long periods of time means that they can be stolen and used by fraudulent users. Human memory limitations often exacerbate these issues, causing people to choose weaker, easy to remember passwords and to use the same passwords across multiple services, allowing a data breach from the least secure service to compromise the user's accounts on all services sharing a password.

A method of increasing authentication security that is growing increasingly common is Multi-Factor Authentication (MFA), in which two or more types of authentication are used at once, so as to increase overall security. In order to compensate for the lacklustre security of password-based authentication, other authentication frameworks are being created to help create overall more secure authentication systems.

We believe that a new authentication framework utilizing 'something you do' and 'something you know' mechanisms has the potential to be a very valuable element of a MFA system, used in combination with other mechanisms to dramatically increase security while still being accessible and convenient. There is a new authentication framework posed by Dr. Ouda which seeks to accomplish this by means of observing user behaviour and constructing authentication questions from the observed behaviour, which will be easy for the authentic user to remember and answer and hard for a fraudulent user to guess.

Dr. Ouda's proposed authentication framework can be divided into four components, as shown in Figure 1.1. The first phase of the authentication framework uses Big Data techniques to gather large amounts of data for use in the rest of the framework. The second phase, known as Data Security-Based Analytics, uses a variety of statistical techniques to detect anomalous events from within the gathered data. The third phase, known as Human Dynamics Insight And Metrics, leverages the detected anomalies to create a useful authentication system. The last phase, known as Big Data-Driven Authentication As A Service, integrates all of the prior steps into a single software that can be used in MFA applications to increase the security of various authentication systems around the world.

The portion of this authentication framework focused on in this thesis is known as JitHDA (Just-in-time human dynamics based authentication engine) and refers to a mechanism that takes observed user behaviour and constructs meaningful authentication questions out of them in order to create a chatbot-based authentication mechanism utilizing ’something you do’ and ’something you know’ authentication styles.

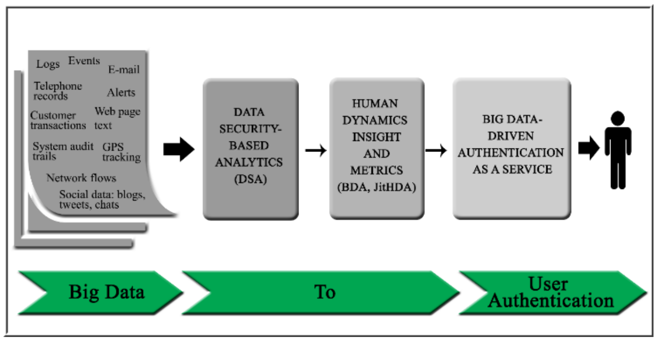

Logs
Events
E-mail
Telephone records
Alerts
Customer transactions
Web page text
System audit trails
GPS tracking
Network flows
Social data: blogs, tweets, chats
DATA SECURITY-BASED ANALYTICS (DSA)
HUMAN DYNAMICS INSIGHT AND METRICS (BDA, JitHDA)
BIG DATA-DRIVEN AUTHENTICATION AS A SERVICE
Big Data
To
User Authentication

Figure 1.1: The components of Dr. Ouda's authentication framework[19]

1.2 Research Objectives

The primary goal of this research is to design an initial implementation of JitHDA, which we call Autonomous Inquiry-based Authentication Chatbot (AIAC), that provides some critical components which implement several important requirements of Dr. Ouda's authentication framework[19]. Accordingly, this design should be practical and viable with modern technology and authentication paradigms, while also properly fulfilling the end goal of accurately discerning between authentic and fraudulent users.

The objectives that we pursue in order to achieve that goal are as follows:

We analyse the current state of the art in authentication technology to understand whether or not our contribution has already been made or otherwise rendered redundant.

We provide an initial design that strives to fulfil some of the main functional components of the desired functionalities of JitHDA.

We create a few prototype tests that demonstrate key functionalities of the system in controlled environments so as to demonstrate that the functionalities can be expected to be viable to implement in a real-world system.

1.3 Research Methodology

The first goal is to determine the uniqueness of the topic of this research, and it can be subdivided into the following steps:

i) Determine what overall functionality the system designed in this research will have.

ii) Examine the state of the art to find all recent developments in authentication or chatbot technologies.

iii) Validate that the state of the art does not include the functionality and design that this research intends to cover.

The second goal is the designing of the mechanics of AIAC, and it can be subdivided into the following steps:

i) Planning out a mechanism by which AIAC can accurately select useful anomalies from externally-provided anomaly profiles.

ii) Planning out a mechanism to transform a given anomaly into a coherent authentication challenge that the user can understand and answer.

iii) Planning out a mechanism by which the user's response can be compared against the expected answer to determine how close to correct the response is.

The third goal is the creation of prototype tests for various key functionalities of AIAC, which can be subdivided into the following steps:

i) Construct an experiment that uses a neural network and labelled anomaly data to estimate expected utility of anomalies, manually create a suitable dataset containing simulated anomaly answers, and then observe how well the neural network can learn from the dataset.

ii) Construct an experiment that uses a neural network and labelled anomaly data to rank authentication challenge templates by comprehensibility for a given anomaly, manually create a suitable dataset containing simulated comprehensibility answers, and then observe how well the neural network can learn from the dataset.

iii) Construct an experiment that compares two arbitrary words to create a granular similarity score, create a suitable dataset of words with varying levels of similarity, and then test the experiment on the dataset and ensure it performs with sufficient granularity.

1.4 Research Contribution

Dr. Ouda's new authentication framework, shown in Figure 1.1, seeks to use the principles of Big Data and Machine Learning to create a new authentication system for use in MFA applications. The third phase of Dr. Ouda's new authentication framework, called Human Dynamics Insight And Metrics, contains a mechanism called JitHDA which uses data from previous phases as the input of a chatbot-based authentication engine. This thesis provides the design for an implementation of JitHDA, in order to help complete the implementation of Dr. Ouda's new authentication framework.

Our contribution begins with a literature review examining cutting edge advances in the field of authentication, chatbot, and natural language understanding technologies. With this we conclusively show that such a project as AIAC is novel in these fields and can stand as an advancement of the frontier of human knowledge.

This paper discusses the necessary requirements of AIAC and examines the various means by which it can be accomplished. This includes the features pertaining to the handling of the input data, the features pertaining to the chatbot structure, the features pertaining to the natural language understanding mechanisms, and the features pertaining to the machine learning applications, among others.

We then, for each requirement of AIAC, chose the best means to implement it and designed the mechanism of that segment of AIAC in full detail, thus in aggregate creating a full detailed design of AIAC.

In order to demonstrate that these ideas can be expected to be viable in practice, we design and run three experiments to realistically emulate the conditions of the three most key mechanisms in AIAC, showing that all of the ideas used to create AIAC are likely to be applicable in practice.

In summation, the contribution of this paper is the coherent and detailed design of an authentication chatbot based on natural language understanding technologies and capable of self-improvement, created to implement the JitHDA component of the Human Dynamics Insight And Metrics portion of Dr. Ouda's new authentication framework and serve as a new secure method of authentication.

1.5 Research Outline

The thesis structure is ordered as follows. Chapter 2 shows a literature review of relevant techniques in chatbot technologies and natural language understanding. Chapter 3 describes the proposed design of Autonomous Inquiry-based Authentication Chatbot (AIAC), the system that implements JitHDA. Chapter 4 details the expected implementation of AIAC and demonstrates that the implementation would be possible to create. Chapter 5 delineates how much of Ouda's authentication framework is already designed and implemented and how much design and implementation is specific to this thesis. Chapter 6 concludes the thesis and discusses what more can be done on this topic in addition to what the thesis itself has covered.

Chapter 2

Literature Review

This chapter conducts a review of current works related to our relevant subjects, such as advancements in authentication systems or Natural Language Understanding. In doing so, we also seek to elaborate on the underlying concepts of the field so as to demonstrate the capabilities and limitations thereof, allowing us to in turn show the novelty of our idea and its place within the existing corpus of scientific literature.

For context, there are three primary domains of research that this thesis exercises: chatbot software, natural language understanding, and authentication. Chatbot software refers to a type of application in which the user and the program exchange text output back and forth for a variety of purposes, such as automated customer support, data collection, or entertainment. Chatbot technology has existed for a while but is starting to find many new applications as modern technology advances and is becoming a promising new area of study.

Natural language understanding is an application of machine learning, which refers to a software paradigm in which, rather than execute pre-specified rules of operation, a piece of software is equipped with the ability to take in data and use it to reshape its operational behaviour, thus ’learning’ from the data. Machine learning is a very exciting new field with many applications being developed to create programs that can automatically learn things that are difficult to manually specify, creating a general increase in software capabilities as difficult tasks can be performed easier and faster than ever before.

Authentication is the practice of limiting access to a service or resource behind some identity-determining criteria. Authentication applications serve to gather information from the user in order to determine if they are one of the people who has permission to access the given service or resource, and range from password technologies to biometric scans and everything in-between.

In the field of chatbot technologies, we will now examine new applications of chatbots and associated developments so as to gain an understanding of the current state of the field and where it might progress in the near future.

Debmalya Biswas[3] describes the growing security concerns inherent in chatbot programs, due to the potential for Natural Langauge Understanding technology to be used to take advantage of user information revealed in chatbot conversations and due to the inability to keep logs completely private so that they can be used to help train chatbot programs. Two strategies are suggested to mitigate this concern. The first of the two automatically detects any statements that might reveal personal information unnecessary to the query itself, such as location or sentiment, and strips those qualities out of the query before sending it to the chatbot server. In the case where the structure of the chatbot is not known, a separate algorithm can be used to reduce unnecessarily-given information by extracting key information from the message and comparing it against public qualities of the chatbot, only sending the full message if it can verify that the message is applicable to the scope of the chatbot.

V Akshatha Prasad et al.[21] proposes a means of authenticating individuals by recognizing patterns in their voice. Features are extracted from audio clips of the given user speaking, and compared against stored features of the authentic user from a calibration period using Vector quantization (VQ) matching. This technology further allows for the parsing of commands by authenticated users to perform functions such as turning lights on and off. The overall success rate of the matching algorithm in correctly matching audio clips with their corresponding user is 71.42%, which is promising but not high enough to be practically useful. This method of authentication stands out as a way to, in theory, utilize biological characteristics of the user to conveniently and securely authenticate users. However, in addition to the immature strength of the technology, it relies on high-quality audio input, which makes authentication relatively inaccessible.

P. Srivastava et al.[24] describes a chatbot which can automatically diagnose a person's medical troubles by means of a short conversation in which it continually gathers information, ruling out ailments until it has a shortlist of possible illnesses and eventually a most probable illness. Once it has confidence about what the user is suffering from, it can begin automatically recommending advice on what the user should do to deal with the problem.

B. Liu et al.[14] analyses user statements to build up a user profile that helps chatbots personalize interaction with the user. Drawing on historical context, a two-branch neural network is used to compare the model of the user both with the provided user post and a candidate response in order to determine the suitability of the candidate response, with the results updating the user model for future conversational fine-tuning.

C. Kao et al.[9] creates a model for chatbots to display emotions in response to the user behaviour. It includes a mechanism to analyse user input and determine the emotions present in the sentences, and uses that to determine a suitable output emotion to affect the output text. This helps serve the objective of creating chatbots that accurately reflect the emotional demeanour of human beings for the purpose of more natural and intuitive communication with users.

F. Patel et al.[20] seeks to determine if a user is feeling stressed or depressed by analysing chat text from them, for the purposes of identifying how the chatbot can help the user maintain a healthy mental state. It uses a Convolutional Neural Network, a Recurrent Neural Network, and a Hierarchical Attention Network as the possible methods to build a profile on the user's emotional state. This demonstrates the increasing capability of chatbot software to engage in complex and subtle analysis in order to add nuance to interactions and create new opportunities in which chatbot software can be productive and useful.

N. T. Thomas[26] details a method of parsing user input without machine learning, using instead Artificial Intelligence Markdown Language (AIML) and Latent Semantic Analysis (LSA). The AIML phase of operation relies on manual configuration creating connections between certain inputs and certain FAQ pages, while if that fails the LSA phase compares the input against the FAQ pages to choose the FAQ page with the most in common with the input.

A. Das et al.[5] combine a chatbot structure with computer vision to answer questions about a picture in natural language. The context of their data collection is one person attempting to explain an image to another person, in which the person who cannot see the image asks questions of it and the person who can see the image answers the questions. This paradigm reduces bias with respect to the nature of the questions the questioner asks (as they do not see specific features to ask about) and increases the frequency of binary questions (questions which can be answered with 'yes' or 'no'). In terms of answers the dataset created contains more long answers, and also includes more questions with ambiguous responses like ’I’m not sure’ as a consequence of the questioner not being able to see the picture and choose answerable questions. Furthermore, of binary answers in this dataset, more of them are ’no’ than ’yes’ while previous similar datasets had a reversed trend, also a consequence of the questioner not being able to see the image.

Also important for the program to function is an understanding of pronouns (as most conversations utilize pronouns at some point) and temporal flow (as people frequently stay on one topic for more than one question). As it is difficult to evaluate how natural and coherent an output is, instead of generating natural output the algorithm sorts a list of 100 candidate answers from most to least appropriate. The candidate answers are drawn from the ground truth answer, answers to similar questions, common answers regardless of question similarity, and a few entirely random answers.

The algorithm uses three encoders to transform the input available to the chatbot into a vector space. In all cases, the image input is handled by the VGG-16 Convolutional Neural Network (CNN) which translates a static image to a series of abstract features for machine learning uses. The first encoder is the Late Fusion (LF) Encoder, which separately encodes the question and the previous events with Long Short-Term Memory (LSTM), a method of holding information in machine learning algorithms for future use in the algorithm, and concatenates the outputs vectors together. The second encoder is the Hierarchical Recurrent Encoder (HRE), which embeds the image and question together with LSTM (early fusion), embeds each round of the history, and passes the concatenated vector to an RNN which produces an encoding for this round and a dialogue context for the next round. The third is a Memory Network (MN) Encoder which treats every previous question as a ’fact’ and learns how to refer back to those facts and the image to answer the question.

The algorithm also has two decoders to utilize the vector space to rank the candidate answers. The first is the Generative decoder, which utilizes LSTM and is trained solely to maximize guessing the ground truth as the highest answer, as while training by full ranking would allow the algorithm to exploit biases in candidate question generation, that might not indicate real progress and distract from actually determining the ground truth. The second is the Discriminative decoder, which uses a softmax function that normalizes input data into a probability distribution which can be trained to optimize the posterior probability of the correct answer. As with the Generative decoder, in training the Discriminative decoder is only evaluated on the likelihood of guessing the ground truth.

This type of chatbot problem is comparatively novel, but demonstrates significant accomplishment in the assembly of a chatbot system that can meaningfully understand and answer freeform questions about a specified subject matter.

Next, in the field of natural language understanding technologies, we can see what the state of the art looks like by examining notable recent papers.

Dena Mujtaba et al.[17] provides a general overview of the state of NLU technologies as it pertains to procedural knowledge, which can be understood as knowledge of steps (often multiple) in service of an ultimate goal, such as a recipe. It outlines the various tasks that compose procedural knowledge understanding, and groups them into three categories: Information acquisition, Information extraction, and Knowledge representation. In describing these categories, it refers to many other recent works that relate to these tasks, forming a hub of knowledge as it pertains to the tasks necessary for procedural knowledge understanding.

Eunah Cho et al.[4] tackles the NLU challenge of semi-supervised learning, in which a

NLU system trained with labelled data is able to examine unlabelled data to update its models automatically. To improve on extant such methods, they create a system that optimizes for diversity in the unlabelled dataset in addition to the typical optimization of model performance, under the idea that unlabelled data typically holds low diversity (i.e. many values are very similar to each other) and thus that an algorithm to select high-diversity data out of the unlabelled dataset can create a smaller dataset of unlabelled data to integrate into the model while retraining model performance.

B. Rychalska et al.[22] is an endeavour focused on analysis of user input statements, in order to help understand complex input statements with multiple intents. It contrasts and implements multiple approaches of breaking multi-intent sentences into single-intent components, demonstrating the effectiveness of the method.

Y. Lan et al.[13] involves using an input question to search a graph for nodes and connections that may be relevant to the question. It draws attention to the creation of multiple points to start searching by using external resources to identify other things similar to the input, and to its use of a 'matching-aggregation' framework to help determine the final prediction.

K. J. Jose et al.[8] present an adjustment to the way slots are parsed from natural language input. The typical method of parsing statements is to assume only one relevant intent with all identified keywords intrinsically relating to that intent. Instead, this paper constructs a data structure in which keywords are stored with association to an intent, allowing for datasets of sentences with multiple intents that are still comprehensible to neural network predictive structures.

R. Kulkarni et al.[11] creates a structure called a 'Calibrated Quantum Mesh', a type of graph built to understand language the same way humans do. It is capable of associating multiple meanings to the same word and interconnecting meanings based on calibrations from non-direct unannotated training data. The purpose of this technology is to effectively find appropriate FAQ pages for a user input.

B. Setiaji et al.[23] details a few specific functions that can be used during the process of converting natural language to a format suitable for machine learning applications. It then describes a usage of the converted words involving Artificial intelligence markdown language (AIML) to match the input strings to specific responses.

Lally, A et al.[12] present a software that builds upon the extant question-answering application Watson in order to permit the solution of more complex scenario-based questions where the answer may be a composite of facts drawn from multiple sources and general knowledge.

In order to answer such a question, the first step is to decompose the scenario into multiple simple questions that the regular Watson algorithm could solve. This is done by identifying keywords and other significant factors in the scenario and assembling an 'assertion graph' out of them to work with for the rest of the algorithm. Since most scenarios will have more data than is likely to be worth trawling through, the next step is 'node prioritization' in which the most useful and relevant of the assertion graph (such as tests with abnormal results) are weighted stronger than other nodes.

Watson is asked a variety of questions based on the assertion graph, with queries constructed from one or more nodes of the assertion graph. All the relevant data from Watson's output, including all of the top answers and confidence metrics, are added to the assertion graph as nodes connected to the nodes their question was derived from.

After the questions have been asked, node weights are recalculated using probabilistic inference methods, and then the system iterates again by asking more questions to Watson from the expanded graph. At the end of each iteration a check is made to see if any generated hypotheses have become particularly highly-weighted.

WatsonPaths was tested on 2190 medical test preparation questions (filtered such that incompatible questions that require image recognition or do not accept text segment answers are not included), reaching 48.0% accuracy on the full set and 64.1% accuracy on the subset dealing only with diagnosis questions. This noticeably outperforms regular Watson's results on the same dataset (42.0% and 53.8% respectively).

Lastly, we can examine recent advances in the general principles of authentication in order to build an understanding of what place authentication has in our modern world and what developments are beginning to take shape.

T. Zhu et al.[30] presents a way to improve authentication on mobile devices by warding against the threat of someone's phone getting into the wrong hands. It discusses the option of using various biometric readings gathered from the phone sensors to compare against known patterns for the user, allowing the phone to recognize when someone else has taken the phone through means such as motion sensors measuring the holder's gait. This will make for a useful 'something you are' authentication as a person's gait is tied to their physical characteristics and is very hard to fake for other people with different physical characteristics.

S. Kim et al.[10] proposed an extension to the SASL (Simple Authentication and Security Layer) authentication framework that allows the user to select various authentication levels with various permissions after authentication. This allows people to customize what level of security they make use of based on their current needs for the application, rather than accessing unlimited access every time. This serves as a way to mitigate the tradeoffs between secure and accessible authentication, providing a framework to assign more accessible authentication methods to the least important levels of authentication but more secure authentication methods to the most important levels of authentication.

L. Dostálek et al.[6] creates a structure for dynamically changing the required authentication method in response to suspicious behaviour or hostile attacks. Authentication methods can be rated based on a set of metrics, including whether the user has unlimited retries or whether it's possible to eavesdrop on the authentication. By rating authentication methods this way, it becomes possible to respond to suspicious behaviour by merely selecting a better-rating authentication mechanism. This serves as another tool in the toolkit of designing authentication methods well-suited to a given service. If authentication methods can be qualitatively rated and compared against each other, it can be made clear which methods are optimal for the situation's needs.

T. Tuna et al.[27] describes a method of performing in-depth examination of social media content in order to discern useful information about a given user. It discusses a variety of example features, such as gender, geolocation, and profession using a variety of methods. With this information, this paper creates a model able to understand social media users well enough to estimate useful metrics such as their expected future behaviour, which can be used for marketing purposes, or their risk of radicalization. It also uses this model to form a categorization system that allows for automatic detection of spammers and bot accounts.

S. Medileha et al.[15] discuss the state of data security in the emerging Internet of Things (IoT) technological environment. The inherent limitations of IoT make it vulnerable to many avenues of attack, including notably a lack of processing power needed for many traditional security algorithms. To solve this problem, the paper creates an Ultra Lightweight encryption algorithm exclusively using operations suitable for low-resource hardware. Using a Raspberry

Pi as example of a low-resource system, the algorithm is implemented and tested against comparable symmetric encryption techniques, showing favourable results.

A. Oracevic et al.[18] surveys a variety of papers on IoT security in order to establish where the state of the art in the field lies. It defines the security considerations that it is taking into account, and examines the surveyed papers through that lens. It finds that many papers on IoT security only focus on one level of the IoT architecture and that there is little work done on ensuring security across the entire IoT architecture based on the standards they defined.

P. Voege et al.[28] examine the state of the art in authentication, chatbot, and Natural Language Understanding technologies for the purpose of ascertaining the novelty of a new authentication system designed to leverage big data to question users on anomalous events in their recent history as the method of authentication. It examined various studies in the relevant fields and found nothing directly applicable to the stated goal, thus proving that said goal is novel in the relevant fields.

P. Voege et al.[29] design and prove the validity of an authentication system sufficient to fulfil JitHDA (Just-in-time human dynamics based authentication engine) by means of converting anomalous events in the user's recent history to dynamically create comprehensible authentication challenges for which the given answer can be accurately rated to determine the authenticity of the user. Multiple experiments are conducted on key components of the design in order to prove that, in a full implementation of the system, its core functionalities will operate as expected and produce reliable outcomes.

Chapter 3

Autonomous Inquiry-based Authentication Chatbot (AIAC)

During authentication sessions, AIAC is designed to interact with the user via a chatbot interface, asking questions and receiving responses until either allowing the user to proceed or denying the user access based on the content of the responses given. The overall goal for how this is to be accomplished is to query the user about details of anomalous events in their recent history. Anomalous events are more easily remembered than normal events, making it more likely that an authentic user would remember the details of the event and be able to accurately answer the question.

AIAC can be broken down into multiple phases of operation: gathering the anomaly profiles, selecting an anomaly to use, creating an authentication challenge out of said anomaly, interacting with the user via the chatbot interface, and then analysing the user response in order to draw a conclusion. Once the conclusion is reached, AIAC will either accept or deny the user and halt operation there.

3.1 Anomaly Profiles

The first crucial step in designing such a system is understanding the nature of the data available for the task, as whatever methods we wish to use to create the desired outputs must be compatible with our inputs.

AIAC operates through the use of anomaly profiles: user-specific collections of anomalous events in the user's recent history. Through Big Data practices large amounts of data can be gathered about each user, and the unusual or unexpected events within that dataset can be automatically classified as anomalous for our use.

The value of using only anomalous data instead of every data point gathered is twofold. First, anomalous data does not predictably repeat, which makes AIAC resistant against attacks in which authentication data is stolen from one session and used to gain entry in a later session. Second, anomalous data is memorable, representing events that stand out from the background routine of the user's life and thus are more easily remembered than ordinary events. While it is also easy to remember some routine events, those are already precluded under the first reason.

The source of our input data is not within the scope of AIAC or this thesis. The gathering of the data is handled by external tools, and the mechanism by which anomalous data points have been isolated has already been designed and implemented in the scope of another project[1][25]. This other project identifies anomalous data within a dataset by automatically organizing the user data by all of its features and then discerning trends within the data to isolate outlier data points which do not match the trends.

Even though the data gathering and anomaly detection is outside the scope of our project, it is important for us to properly understand the structure of our input data. The data is segregated by user, such that each user has their own anomaly profile based on their recent commercial transactions. Anomaly profiles only contain recent transactions because they are of a finite size and, if they are full and a new anomaly is introduced, they will remove the oldest anomaly in the profile to make room. This means that the information used to make the user's authentication questions has an intrinsic time limit, such that if any malicious attacker got their hands on authentication-relevant information it would automatically become inapplicable after a certain amount of time. This can be compared to regular password resetting, except inherent to the functioning of the system and requiring no effort from the user.

In the anomaly profile, after removing features which will not be useful in any step of AIAC's operation, each entry has the following features: The user's unique ID. The type of action the user is recorded as having performed, organized into multiple categories depending on the nature of the actions performed. The date on which the action was recorded. A record of which quality of the event was found to be anomalous.

Two more features can be derived from the anomaly profile: The exact value of the feature that was found to be anomalous. The value expected for said anomalous feature.

Utilizing all of these features, it should be possible to create a clear model of the situation and context in which the anomalous event happened, giving AIAC the ability to generate useful authentication challenges out of it.

3.2 Selecting Anomalies

The first major task we needed to accomplish is the selection of an anomaly to present to the user. Given the externally-provided input of an unsorted list of possible anomalies, the problem can be rephrased as sorting the anomaly list from most to least desirable for use in subsequent steps.

The goal of this section is to create a method to determine whether or not a given anomaly will make for a useful authentication challenge, meaning that we want the anomaly to help us discern between authentic and fraudulent users. The outline of the method we created to accomplish this is as follows: we train a neural network on automatically-collected data from the performance of past anomalies, and then use the neural network's model of anomaly quality to rate new anomalies as they are introduced, allowing for fast and accurate selection of useful anomalies.

Neural networks, shown in Figure 3.1, are an application of machine learning, a paradigm in which computer programs take in data to improve their own intelligence and performance so as to act with intelligence on their designated subject matter. A very common example is programs that predict trends within datasets, automatically creating a model out of the data to extrapolate or interpolate from.

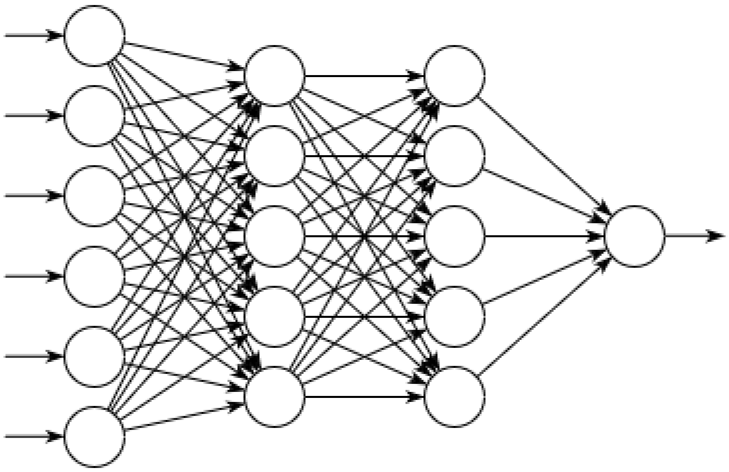


Figure 3.1: A feedforward neural network with one output

In our solution to the problem of selecting useful anomalies, we seek to create a neural network that automatically generates a model of what makes anomalies useful, sparing us the need to discern the many and subtle rules by ourselves. The only thing we need to make this a reality is sufficient quantities of input data, which brings us to the other half of our strategy.

The only way to generate truly accurate data on whether an anomaly is useful for authentication purposes is to empirically test it. A useful anomaly will be correctly answered by an authentic person and incorrectly answered by a fraudulent person, and vice versa for maximally useless anomalies. By recording how the user responded to the anomaly and contrasting it against what we know of the user, we can determine just how useful the anomaly was and use that information to help guide future anomaly selection.

By assembling a large amount of this data through empirical use of the program, the final part of our anomaly selection strategy can be employed, using a neural network to learn the trends of useful anomalies and gain the ability to better predict future anomalies. These abilities will grow only more accurate and precise the more the system operates, as each new interaction will generate some new information to add to the training dataset. In addition, our strategy can automatically future-proof itself by using the new information to adapt to changing contextual conditions, adjusting its predictions as societal context changes and new trends appear and old trends vanish.

3.3 Forming Authentication Challenges

The next major phase of the project is to devise a mechanism by which a chosen anomaly can be transformed into a chatbot-suitable authentication challenge

We can, if there is any utility in it, use the features of the original value of the anomaly that were removed in the preprocessing of the anomaly selection.

The technology exists to dynamically create sentences in natural language with machine learning algorithms, and such technology would be capable of taking the information of an anomaly and converting it into a suitable authentication query, but at the same time such technologies cannot guarantee that their output will be coherent, only that their output is likely to be coherent. As a result of this downside, we chose to design a different means of authentication challenge generation specifically for this problem.

In a world in which we only had one kind of question to ask, such as how much the user paid for a grocery purchase, we could guarantee a coherent question by simply using the fixed question of 'how much did you pay for groceries today?'. Building off of this principle, if we have a finite number of scenarios, we can create a finite set of predetermined questions to choose from and choose which question to use based on the details of the anomaly.

This mechanism can be expanded by making the predetermined question flexible to a limited degree in accordance with the details of the anomaly. For instance, the date of the anomaly can be used to help phrase the question, allowing us to condense two questions like 'how much did you pay for groceries today?' and 'how much did you pay for groceries yesterday?' into a single statement that uses the date of the anomaly to fill in the word. We can call such structures authentication challenge templates, shown in Figure 3.2, and they are the cornerstone of our strategy for this task.

Following this principle we can draw on many different qualities of the chosen anomaly to create coherent and precise authentication challenges with minimum effort. Given that different anomalies contain different kinds of information, for a given anomaly we can expect some authentication challenge templates to be ineligible, expecting information only available in

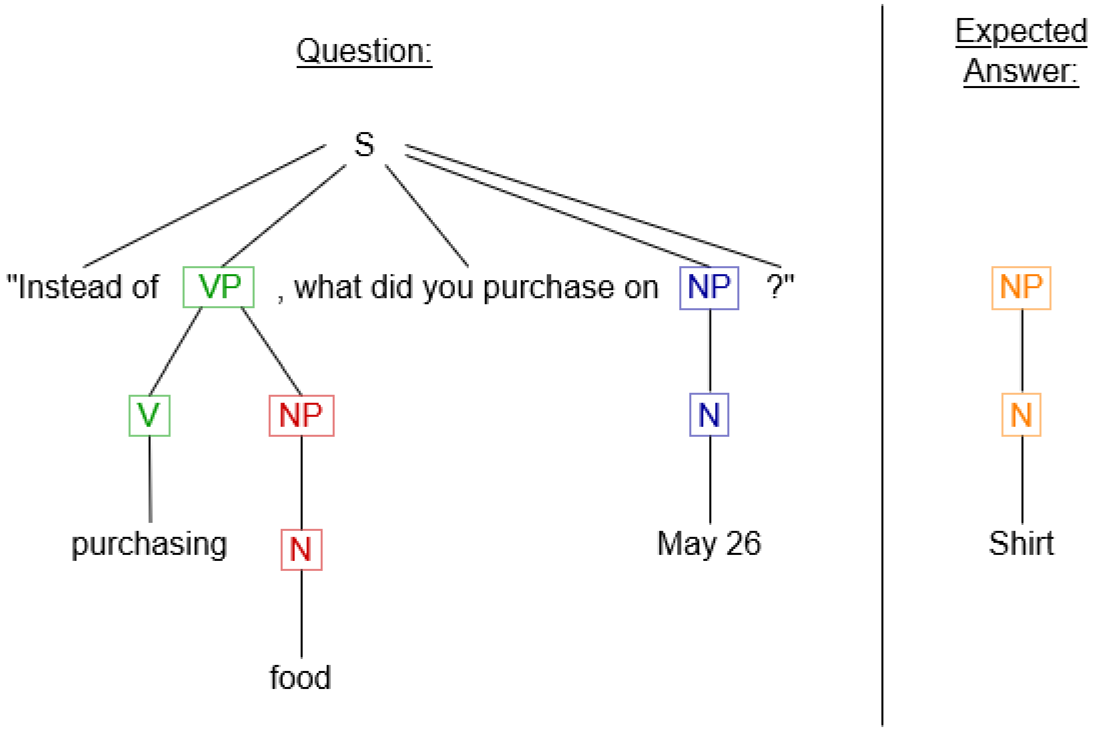


Figure 3.2: An authentication challenge template

different types of anomalies. We can institute some basic logic to exclude such authentication challenge templates from consideration automatically.

But beyond this, we may also find ourselves wanting a variety of authentication challenge templates, even when drawing the same information out of the anomaly. An authentication challenge is composed of a question and an answer, each utilizing different information from the anomaly, and you can create meaningfully different authentication challenges simply by modifying which information from the anomaly is used to form the question and which information is used to form the answer. Therefore, while we may able to exclude any authentication challenge templates that require information not provided by the given anomaly, we may still find ourselves faced with a choice between multiple possible authentication challenge templates to use for the anomaly, and thus we need a selection mechanism to choose the one most suited to our purposes.

While the use of authentication challenge templates guarantee comprehensibility to a certain degree, we can extend our strategy to pursue even greater comprehensibility. Despite the fact that all authentication challenges will be semantically accurate, there is no guarantee that they will make sense as a scenario. If, for every anomaly, there is a selection of authentication challenge templates that will have varying comprehensibility, we can create a mechanism to try and optimize for the most comprehensible authentication challenge template.

Since ’comprehensibility’ is a difficult concept to define programmatically given how much complexity exists in human languages, attempting to evaluate comprehensibility with a direct algorithm would pose many challenges. Instead, we looked for an alternate approach to optimizing for comprehensibility.

The key to this part of our strategy is user behaviour during the authentication sessions. We already draw on user behaviour to determine whether an anomaly was of help in determining the user’s status as authentic or fraudulent, but there remains another aspect of the feedback that we can make use of. In the event that the presented authentication challenge is not very comprehensible, it is probable that the user will react with confusion, which can be detected from their response. Therefore, when the system detects confusion in the user response, we can determine that the pairing of that anomaly and that authentication challenge template was not very clear. Similarly, if the user does not express confusion, and instead attempts an answer, then regardless of what the answer may be we can determine that the pairing of that anomaly and that authentication challenge template was comprehensible.

Accordingly, we use this strategy to create labelled data depicting the comprehensibility of various pairings of anomaly and authentication challenge template. This labelled data empirically depicts the dynamics of authentication challenge comprehensibility in a way that can theoretically be understood by a computer program. The last part of our strategy is making use of this data and, like with determining which anomaly to use in the first place, a neural network is a good choice for this task.

However, there are many authentication templates to choose from, and only one authentication template referred to in each piece of labelled data. This means that instead of the neural network outputting only one value, it needs a number of outputs equal to the number of authentication templates coded into the system, so that each anomaly can be compared with every authentication template available, allowing for the most comprehensible pairing to be selected. The structure of a neural network with this type of output can be seen in Figure 3.3. As with anomaly selection, this system will improve its capabilities over time as more and more comprehensibility data is collected to fine-tune the network.

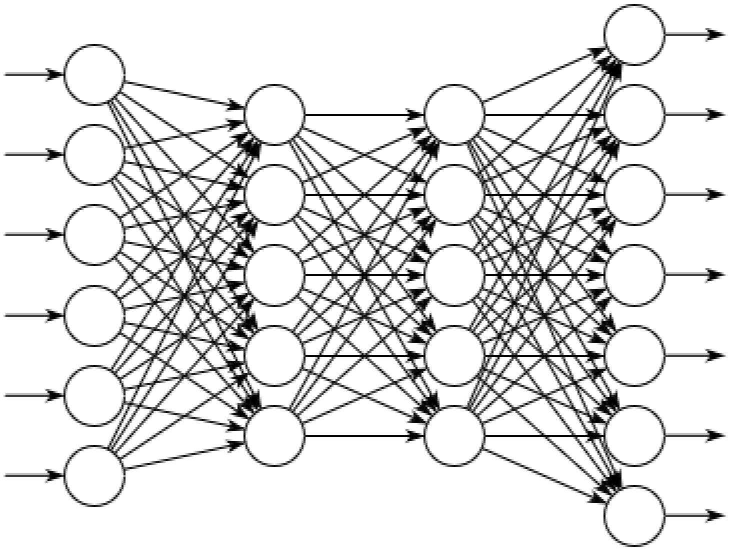


Figure 3.3: A feedforward neural network with many outputs

3.4 User Interaction

Once an authentication template has been selected, there is only a little more work before it is presented to the user. Following the indicators in the template for what sort of data is needed in which spots, we can draw data from the anomaly in question to fill out the authentication challenge template with user-specific information, creating an authentication challenge with two pieces of data: a question in the form of a string, to be presented to the user, and the expected answer in whichever format is specified by the template.

When the user attempts to log in to a given user account, they will be presented with one of these authentication challenges as a prompt on their computer in a chatbot format. There will then be an input field for the user to give their response to the question. Once the user inputs their response and submits it, the program can proceed to analysing if the user response is likely to be authentic or fraudulent.

3.5 Analysis of User Response

Our strategy for the process of analysing the user's response and using it to determine the authenticity of the user can be broken down into two steps. The first step is to check for signs of incomprehension on the part of the user, such that their answer reflects not an accurate or inaccurate answer but a failure of the authentication challenge to accurately convey what information is desired from the user. This may be accomplished by an input method separate to the prompted answer field, such that we can merely read the input from that source to determine user comprehension, but it is also possible to check the prompted input for specific words or phrases that indicate incomprehension, such as ”What?” or ”I don't understand”. If such signs are found no further analysis of the user response is necessary as the validity of the user's answer cannot be measured if the user did not know what they were supposed to answer, and AIAC will proceed to the post-analysis step.

The second step of the analysis process is initiated once it is determined that the user did in fact comprehend the authentication challenge. In this step, AIAC compares how close the user response is to the expected answer. Our strategy for this varies depending on whether or not the expected answer is numeric.

When the answer is a numeric value, such as the amount spent on a given transaction or the time at which it happened, the method of comparison we employ is to attempt tp parse the user's response as a number so as to create a direct numerical relationship between the user response and expected answer. If the user answered with a non-numeric value that nonetheless corresponds to or otherwise indicates a numeric value, we can design mechanisms to recognize such cases and convert them to numeric values in order to create the aforementioned numeric relationship. For instance, we would take the response ”twenty bucks” and interpret it as 20.00 dollars of local currency. We can determine the exact degree of similarity between the two comparable numeric values by using a normalization process based on the entire user profile, such that their common behaviours for the type of value in question are factored into the decision of precisely how far off the user response is from the expected answer.

When the answer is not a numeric value, however, the comparison must be made between two words or phrases. Unlike with numeric values, there is no straightforward way to derive a granular similarity metric between any two words, and so our strategy employs a more complicated process. The principle behind this method is that highly similar words may show up as synonyms of each other, and that this effect might be chained onwards so as to connect less similar words in a chain of synonyms. For instance, ’sneaker’ and ’boot’ might both be described as synonyms to the word ’shoe’, and this through the word shoe we can tell that ’sneaker’ and ’boot’ are related words. In this way we can base the connection between two words in how long of a synonym chain you need to reach one word from another, giving us a level of granularity in our measurements to match the granularity in the numeric comparisons.

Once we have finished analysing this specific response, we next devised a means to translate the results of that authentication challenge into the desired goal of authenticating the user. Beginning from a default state of zero evidence, AIAC updates its estimates of the user with the results of each authentication challenge answered until one of two thresholds is reached. One threshold represents sufficient evidence to confirm the user as authentic which, once crossed, indicates for the system to let the user pass, and the other threshold represents sufficient evidence to confirm the user as fraudulent, indicating for the system to instead reject their authentication request.

Should a given authentication challenge fail to result in either threshold being crossed, AIAC proceeds to initiate another authentication challenge, iterating in this way until either a threshold is crossed, the session is externally interrupted, or too many authentication challenges have been executed within the session.

Regardless of the outcome, AIAC will then store the results of the authentication challenges as according to the previous sections, so as to use the data gathered in the authentication session to enhance AIAC’s future performance.

Chapter 4

AIAC Implementation

Chapter 3 depicted the necessary design of AIAC such that it would be a valid implementation for JitHDA, but it is not wholly sufficient to describe the broad strokes of what the system must accomplish. It is also imperative to delve into the details of the system to prove that such a design is truly feasible. AIAC is not yet an implemented system, but in this chapter we describe the complete design as it would be implemented, and in chapter 5 we will demonstrate through some experimental trials that the design can be expected to work as intended, and thus that the following design of AIAC will be a valid implementation of the needs of JitHDA. The overall process we will describe is modelled using a UML activity diagram as shown in Figure 4.1.

4.1 Anomaly Selection

The focal point of the anomaly selection process is the neural network, as the dynamics of what makes an anomaly 'useful' is subtle and ever-shifting, making it impractical to manually

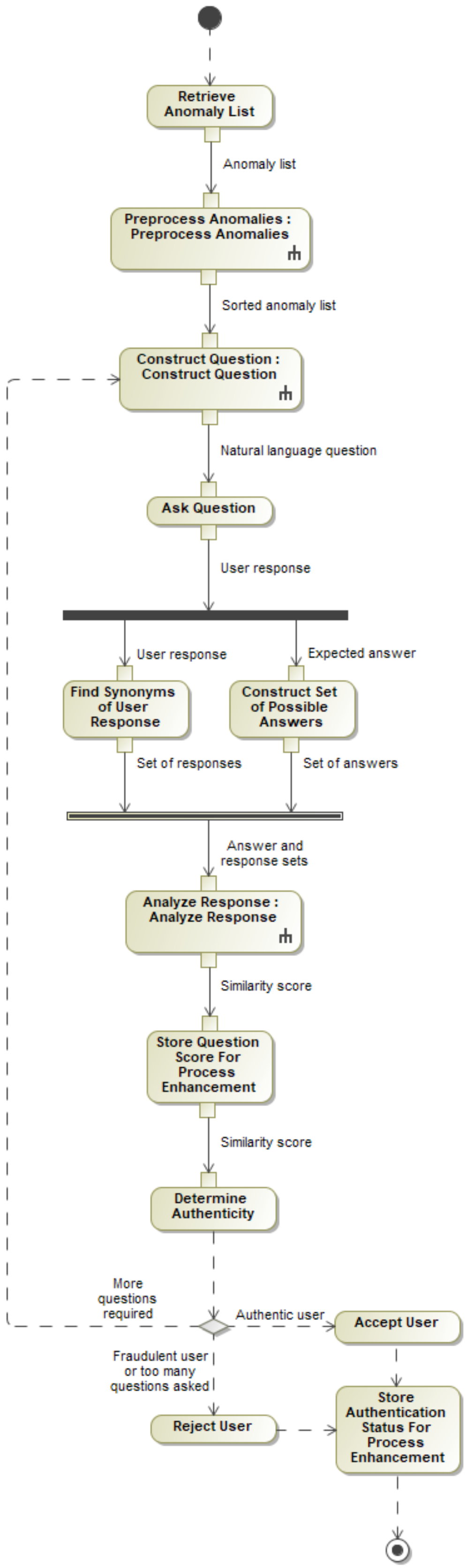

Retrieve Anomaly List
Anomaly list
Preprocess Anomalies : Preprocess Anomalies
Sorted anomaly list
Construct Question : Construct Question
Natural language question
Ask Question
User response
User response
Expected answer
Find Synonyms of User Response
Construct Set of Possible Answers
Set of responses
Set of answers
Answer and response sets
Analyze Response : Analyze Response
Similarity score
Store Question Score For Process Enhancement
Similarity score
Determine Authenticity
More questions required
Authentic user
Accept User
Fraudulent user or too many questions asked
Reject User
Store Authentication Status For Process Enhancement

Figure 4.1: The overall process

derive and hard-code the necessary logic of this decision. Instead, we rely on an automatic and dynamic process for making this decision based on the principles of machine learning. A neural network provides the flexibility to dynamically learn what makes anomalies useful from empirical data, and it has the potential to update its internal logic over time as external conditions change and the old answers for what makes an anomaly useful become less accurate.

The most suitable type of neural network for this type of scenario is a feedforward neural network because we are attempting to map anomalies, which can be represented as a set of numerical features, to a single numeric rating, and the features in question lack any meaningful time-series elements or other qualities that would be better solved by other types of neural networks.

In order to train this neural network, we need labelled data indicating whether given anomalies proved useful or not, which is determined by whether or not they helped the system accurately discern between authentic and fraudulent users. We can determine this by comparing the result received for that anomaly against the final verdict for the user, deciding that the anomalies which were in line with the final verdict were 'useful' and that the anomalies which were not were 'not useful'. In this way, we can trust the utility label to the same degree that we can trust the final verdict.

With the neural network detailed, we can visualize the anomaly selection phase with Figure 4.2.

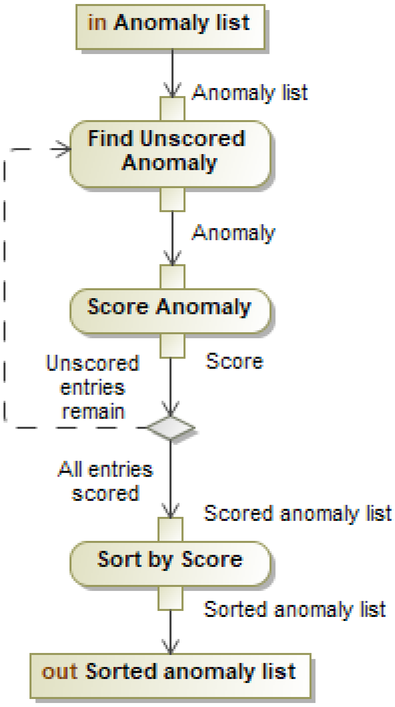


Figure 4.2: Preprocessing the anomaly profile

4.2 Forming Authentication Challenges

The primary difficulty in forming dynamic authentication challenges is in ensuring that they are both flexible enough to handle the entire range of data that the system operates with while never compromising the comprehensibility of the output. We can take as granted that whatever anomaly has been chosen will be useful to AIAC's task, but that is only realized if the user is able to comprehend what is being asked of them.

With anomalies being defined solely as sets of related values about recorded actions, AIAC will itself have to provide the natural language words that the user will be familiar with and able to engage with as a meaningful query. The question then remains of how AIAC will select which natural language words to use, as there is no single satisfactory sentence that will suit every anomaly.

Techniques exist to generate natural language sentences fully dynamically, using natural language processing technologies to take in the details of the anomaly and output a clear question about the details of the anomaly. However, while natural language generation techniques are capable of feats like this to some extent, it is far from trivial to create a system that not only creates comprehensible questions, but comprehensible questions that accurately reflect the core intent of the question being formed.

Instead, we introduce here sentence templates, pre-defined sentences asking questions of various types in various styles, with the keywords replaced with open space for anomaly information to be inserted. When a sentence template is chosen for a given anomaly and AIAC begins to assemble the complete authentication challenge, the sentence template provides information on what exactly is supposed to go within the empty keyword spaces of the sentence. This includes both what information from the anomaly is desired, but also how it should be conjugated or otherwise formatted in order to maintain coherency with the rest of the sentence. In this fashion only simple NLU techniques will be required to form the authentication challenges, once one is chosen for the selected anomaly.

But before any of that can take place, AIAC must first identify, of the sentence templates it is aware of, which of them is best for the selected anomaly. There is some initial screening that can be done, excluding from consideration any sentence templates that require information the anomaly does not have (such as an expected amount of money when the anomaly instead contains an expected activity type), but that alone is insufficient to the task of ensuring that the resulting authentication challenge is as comprehensible as possible. As a result, a second neural network is introduced into AIAC at this step, using the information of the anomaly to more precisely determine which sentence template is most comprehensible with the data of the anomaly.

This neural network is of a similar structure to the one used in selecting the anomaly, except for that instead of one output to rank the anomaly we use one output per sentence template encoded into the system, allowing the neural network to rank all of them by comprehensibility simultaneously. Once that is done, the results can be used to form an ordered list of sentence templates by comprehensibility and select the highest-rated applicable one. The data for training this network can be gained in a similar fashion to the first neural network, by extracting information from the user interactions, but in this case we do not need to care whether the user was authentic or fraudulent in the end, we merely have to care about whether they understood the question or not, which can be evaluated on a question by question basis by looking at whether their response to the question indicates incomprehension or not. In this way we can gradually build up enough data for the neural network to reliably choose the most comprehensible sentence template.

For example, suppose an anomaly is chosen with the following characteristics: (i) Happens on 15/07/2020 (ii) Takes place at a specific Wal-Mart in a specific city (iii) The user purchased new shoes (iv) The nature of the anomaly is the location (v) The expected location value is a specific Old Navy in the same city

The sort of question we would want to see is ”What did you purchase at Wal-Mart on July 15?” with the expected answer being ”shoes”. Because the user typically shops at Old Navy, having purchased shoes at Wal-Mart would stand out to them, and they are likely to remember the event and answer correctly.

The processes detailed in this phase can be visualized in Figure 4.3.

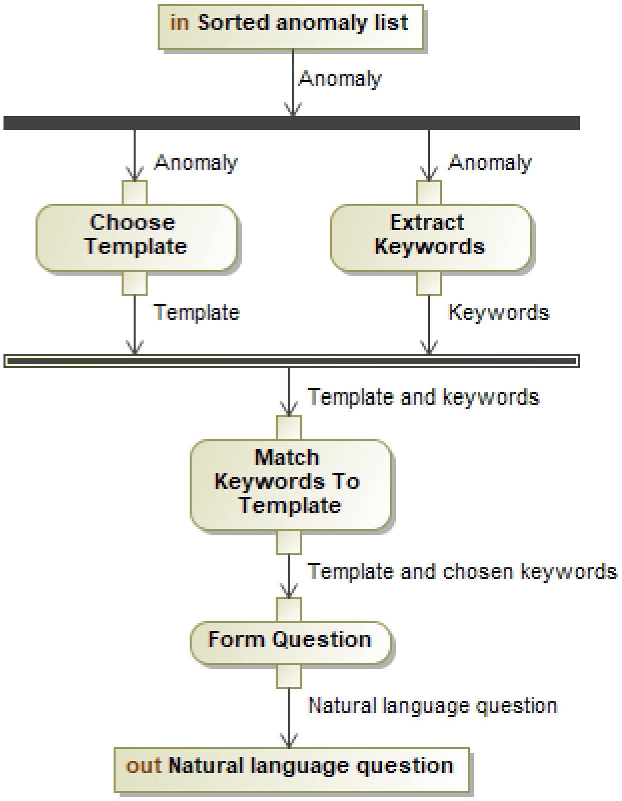


Figure 4.3: Forming a natural language question

4.3 User Interaction

Once we have the complete authentication challenge consisting of a string query for the user and an appropriate data type containing the expected answer, it is relatively straightforward to present this challenge to the user. Once they attempt to log into their profile, AIAC will be activated and generate the authentication challenge for the user to complete. AIAC will be active within a simple chatbot interface, sending text to the user and receiving text responses from the user. After a standardized greeting to explain the broad nature of AIAC insofar as it is relevant to their authentication, AIAC will show the user the string component of the authentication challenge, and wait for the user to respond.

The next line the user inputs into the chatbot system is considered their response to the challenge, which is passed through the rest of AIAC's systems to determine authenticity. AIAC does not give any outward feedback about how the response fared against the expected answer, only notifying the user of their authentication, rejection, or the need for another question, depending on the internal state of AIAC.

4.4 Analysing User Response

With the user response for the posed question collected, AIAC must quickly process the response in order to determine what action to take with respect to the user. The mechanism by which this happens varies based on the data type of the expected response, as some data types allow for simpler and easier comparison methods. The desired output of the first step of this process is a numeric value denoting the similarity between the user response and the expected answer, which can be used to help determine whether or not the user is authentic. If the expected response is a numeric data value, it is possible to compare the similarity between expected answer and user response by attempting to parse the user response as a numeric value and then compare it against the expected answer. To help facilitate this, both numbers can be normalized using parameters derived from information gained from the overall anomaly profile, thus ensuring the derived similarity score is comparable to other normalized similarity scores.

If, however, the expected answer is non-numeric, a different algorithm is used to compare the user response to the expected answer by means of determining the similarity between two phrases. If the user response consists of more than one word, simple word processing algorithms will be employed to reduce the user response to a single keyword suitable for further processing.

Once the user response has been suitably trimmed, the actual comparison mechanism can be used. The principle at hand is that there are a variety of words related to or of similar meanings to the expected answer, and that even if the user does not respond with the exactly correct answer it should be possible to create a measure for how conceptually close their response is to the expected answer, thus allowing us to create a granular measure for how accurate the answer was. This process can be further refined by finding all the words related to or of similar meanings to the user response, and using all of those as candidate words to find a connection to one of the words generated from the expected answer. This process can be iterated further, finding words similar to the words already generated on either side and adding those as candidates to find a match.

If we suppose that ”shoes” is the correct answer to the authentication challenge, and the user provides the response ”sneakers”, we would expect AIAC to conclude that the answer and response are relatively similar, with a moderately high output score. In order to determine this, AIAC would begin by creating lists of synonyms as follows: {1: shoe, 2: boot, 3: cleat, 4: cowboy boot, 5: loafer, 6: pump} and {1: sneaker, 2: cleat, 3: footwear, 4: shoe, 5: tennis shoe}.

Once the set of answers and set of responses have been constructed, they will be examined for any words that appear in both sets. In all such cases, a numeric pair value $\{a, r\}$ is created, with a representing the index of the matched word in the list of answers and r representing the index of the matched word in the list of responses. Thus the numeric pair value encodes how far removed from the ground-truth expected answer and user response the identified word match is.

In the above example, we would find matches between the first entry in the set of answers and the fourth entry in the set of responses, creating the pair $\{1, 4\}$. In addition, we would have a match between the third entry in the set of answers and the second entry in the set of responses, creating the pair $\{3, 2\}$. As a result, $\{1, 4\}$ and $\{3, 2\}$ would be the pair values that AIAC uses to conclude that ”shoes” and ”sneakers” are relatively similar.

Once all numeric pair values (if any) have been found, they will be combined together to create a similarity score. A numeric pair value with lower *a* and *r* values will score higher than a numeric pair with higher *a* and *r* values, and multiple numeric pair values combined will produce a higher score than one alone. AIAC will accomplish this with a sum of reciprocal sums:

$$score = \sum_{i=1}^{n} \frac{1}{a_i + r_i} \tag{4.1}$$

The exact number of iterations desired can be configured based on performance needs and how wide of a net is desired to be cast when searching for potential matches between expected answer and user response. This process of creating similarity scores from the expected answer and user response can be visualized in Figure 4.4, and the specific algorithm used can be described as follows:

The similarity measure created here can then be normalized to the same scales as the the numeric data, allowing them to be used the same way regardless of what type of authentication challenge was asked.

With the similarity measure for the user’s response to the authentication challenge created,

```
Algorithm: Word-Matching Similarity Score
Input  : The expected answer and the user response
Output: A number denoting the similarity between the two inputs
begin
    initialization
    Find n synonyms of the expected answer, retaining listed order from thesaurus.
    Find m synonyms of the user response, retaining listed order from thesaurus.
    foreach input do
        Combine input with synonyms of input into a set of words.
    end
    foreach synonym of expected answer do
        Find n synonyms of the synonym, adding them to the expected answer's list of
          words
    end
    foreach synonym of user response do
        Find m synonyms of the synonym, adding them to the user response's list of
          words
    end
    foreach word in the expected answer's set of words do
        Compare word with each word in the user response's set of words
        foreach match do
            Take the index of the matched word from both sets of words and pair them
              together
        end
    end
    foreach Index pair gathered do
        Sum the two numbers together, then take the reciprocal of the sum
    end
    Sum up every reciprocal for output number
end
```

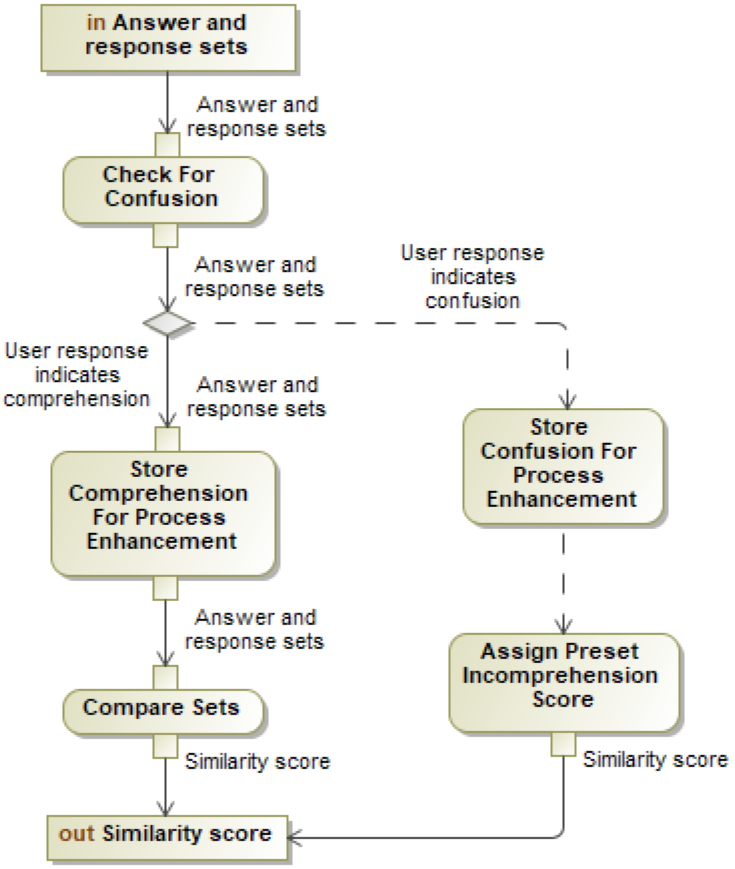

in Answer and response sets
Answer and response sets
Check For Confusion
Answer and response sets
User response indicates confusion
User response indicates comprehension
Answer and response sets
Store Comprehension For Process Enhancement
Store Confusion For Process Enhancement
Answer and response sets
Assign Preset Incomprehension Score
Compare Sets
Similarity score
Similarity score
out Similarity score

Figure 4.4: Analyzing the user's response

AIAC can then proceed to decide what to do with the user. There are three possible outcomes to this algorithm: i) the user is accepted as authentic, ii) the user is rejected as fraudulent, iii) more data is needed to make a conclusive answer. This can be modelled by creating a metric for the user's aggregate performance over all authentication challenges and defining two thresholds on either ends of the scale. If the user's aggregate performance crosses the acceptance threshold, AIAC will proceed to accept them as authentic, and if their aggregate performance crosses the rejection threshold, AIAC will proceed to reject them as fraudulent. At the end of each authentication challenge, AIAC will use the similarity score output to update the user's aggregate performance towards whichever of the two thresholds is more appropriate.

If, however, the similarity score is applied and neither threshold has been crossed, AIAC knows that it needs to pose another authentication challenge in order to gain more data.

Chapter 5

Experiments

To prove the viability of the critical components of AIAC, three experiments have been done, prioritizing the concepts simultaneously most important and most testable. The first experiment pertains to the neural network performing the anomaly selection task, the second experiment pertains to the neural network performing the task of choosing suitable sentence templates for a chosen anomaly, and the third experiment pertains to discerning similarity between two non-identical words, for the purpose of creating a gradient from accurate answer to inaccurate answer in the analysis of the user response.

The primary obstacle to testing these components properly is that, in absence of a fully-implemented AIAC, we are at a dearth of viable data to use in the testing. This is a vital problem for the first two tests in particular, which are meant to demonstrate neural networks learning form appropriate data. In order to rectify this, a modest amount of viable data is manually created in simulation of the expected results of a fully-implemented AIAC. This data is derived from genuine anomaly profile samples, generated externally and provided for use in this project, with the only manual choice involved being what procedures to perform to create the appropriate manual input to the dataset.

5.1 Experiment 1

For the first experiment, regarding anomaly selection, the key behaviour we are testing is the ability to discern useful anomalies, in which 'useful' denotes how likely the anomaly is to procure a correct answer from authentic users and a false answer from fraudulent users. The intended means by which an anomaly would be rated is by comparing how it was answered against the final verdict of the user for that authentication session, such that if a user is determined to be fraudulent any question they successfully answered would be deemed not useful.

In order to appropriately replicate this, each anomaly in the dataset was first assigned an authentication challenge based on the 'anomaly type' feature. Comprehensibility of the authentication challenge here is irrelevant, instead we focus on the data requested by the challenge. To emulate an authentic user, an answer to the challenge can be provided with reference to the other features of the anomaly, and a fraudulent user can be emulated by answering the challenge without reference to the other features of the anomaly. Each anomaly is randomly assigned a label referring to whether the anomaly is to be answered as if by an authentic user or as if by a fraudulent user. As such, we have created a scenario where there are emulated authentic and fraudulent users of known authenticity answering many questions with varying degrees of accuracy, such that answers from emulated authentic users are broadly more accurate than answers from emulated fraudulent users.

In order to derive a feature suitable for the neural network to optimize for, we must compare the answers against the label given. First off, the given response is compared against the expected answer to create some numeric score representing the similarity or lack thereof of the user response. Once this is done for all anomalies in the dataset, the similarity scores are normalized to a z-score based on all anomalies of the same requested answer data type, so that each data type may become comparable to the others. Then, each similarity score is compared against the authenticity label for the anomaly. The more divergent the user response is from the expected answer, the higher the anomaly will be rated if the emulated user is fraudulent, and the more similar the user response is to the expected answer, the higher the anomaly will be rated if the emulated user is authentic.

This is done by taking the authenticity label (0 for authentic, 1 for fraudulent) and subtracting the divergence feature (0 denoting maximum similarity, higher values denoting increasing divergence between expected answer and user response) to create a value representing how close the divergence matched the authenticity of the user, which is our desired feature which we wish to be able to predict so as to identify the most useful anomalies from new, unlabeled data. As such, once we derive this feature across the entire dataset, we move on to the process of training a neural network on the dataset in order to teach it how do make accurate predictions of that feature.

The neural network is a feedforward neural network with one input layer, two deep layers, and one output layer. The size of the input layer is equal to the number of features in each anomaly after all preprocessing has taken place, the sizes of the two deep layers are each twice the size of the input layer, and the size of the output layer is 1, on account of there being only one feature to predict. The deep layers and output layer all use ReLU (Rectified Linear Unit) activation functions, which is a piecewise function that is linear for positive values and zero for negative values. The dataset was randomly split into a training dataset containing 80% of the anomalies and a testing dataset containing the remaining 20% of the anomalies. After being trained on the training dataset, the neural network had an accuracy of 70.8% on predicting the desired feature of the anomaly.

For further analysis of the results, we compared the neural network's predictions to the associated anomaly labels to create an ROC curve, shown in Figure 5.1. It has a smooth curve and an area under curve of 0.96, indicating a healthy amount of separation between the predictions of the authentic and fraudulent anomalies.

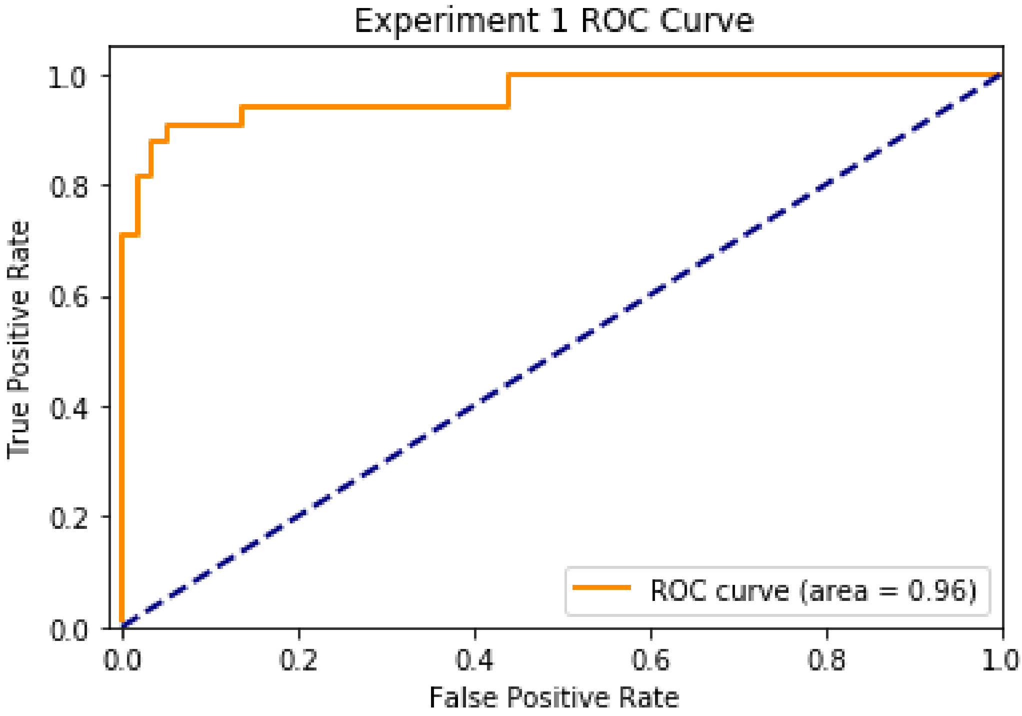


Figure 5.1: Experiment 1 ROC curve

5.2 Experiment 2

For the second experiment, regarding template selection, the key behaviour we are testing is the ability to determine which templates would create legible authentication challenges for a given anomaly, from a fixed set of sample templates. In the intended context of AIAC's use, the legibility of a given anomaly-template pair can be empirically determined by whether or not the user understood what they were being asked, which can itself be determined by analysing their response for words that signify confusion.

The initial phase of this experiment is to create the sample authentication challenge templates that the experiment will be built around. It is important for every type of anomaly to have some authentication template which will provide a minimum of comprehensibility, so that for every possible anomaly there is at least one authentication challenge template for which the result will be a comprehensible authentication challenge.

As we do not have access to a fully-implemented AIAC or the intended context in which it would operate, we must instead replicate via other means the data which would have been gathered. Therefore, each anomaly in the anomaly profile dataset is compared against each authentication challenge template and a comprehensibility rating is manually assigned based on how comprehensible the anomaly is likely to be when applied to said authentication challenge templates.

The method by which the comprehensibility rating in this experiment is determined is primarily focused on the 'anomaly type' feature for the anomaly in question, such that each authentication challenge template has an anomaly type it is geared towards and consequently any other anomaly type would be considered incomprehensible in combination with that anomaly.

It should be noted that such a methodology is a departure from how AIAC would behave in its native context, in which authentication challenge templates may contain any level of specificity or generality, and in which AIAC will learn on its own from genuine user data which authentication challenge templates are most comprehensible to a given anomaly.

The neural network for this experiment is very similar to the neural network for the first experiment. They are alike in using feedforward neural networks with two deep layers, and in using ReLU activation functions, but differ primarily in having an output layer equal in size to the number of authentication challenge templates created for this experiment, as each authentication template needs to receive its own comprehensibility score with respect to the given anomaly.

As with the first experiment, the dataset is randomly divided into a training dataset containing 80% of the anomalies and a testing dataset containing the remaining 20% of the anomalies. After being trained on the training dataset, the neural network had an accuracy of 84.0% at predicting the comprehensibility ratings of each authentication template for a given anomaly.

5.3 Experiment 3

The third experiment pertains to word matching, the process of determining how conceptually close two words are in meaning for the purposes of creating a gradient of accuracy in the analysis of the user response. The principle on which this is accomplished is that two words conceptually close to each other may be synonyms of each other, or share a common synonym, or be synonyms of two separate words which are synonyms to each other, and so on. The principle holds throughout arbitrary degrees of separation, with a weaker effect based on the degree of separation, allowing us to evaluate with granularity how conceptually close two words are.

The example dataset for this experiment does not come from the anomaly profiles, as the anomaly profiles do not provide us with the sorts of information we need to do this experiment, but from a manually-constructed dataset describing words of different conceptual proximity to a base word. In this dataset there are 20 base words, and each base word has one word that is conceptually close to it, one word that is conceptually distant from it, and one word that is wholly unrelated to it. This ensures that we have ample samples for each major gradient of proximity to demonstrate the ability to detect conceptual distance with granularity at every level of separation.

In this experiment, comparison between two words was done with a Python module called PyDictionary, which performs lookups to synonyms.com to return a list of synonyms for a given word. Finding if two words are synonyms is merely a matter of checking if the other word appears in the list of synonyms produced by the first word, or vice versa. We limit the number of synonyms that we accept from any one lookup to the first 10 results, should more than 10 be returned.

In order to evaluate higher-distance comparisons, we take the list of synonyms created from the base word and, using the same PyDictionary lookup, find the synonyms of all the words in that list. We refer to words in the list of words determined by a PyDictionary lookup of the base word as ’first-order synonyms’, and words in the list of words determined by aggregating PyDictionary lookups of all first-order synonyms as ’second-order synonyms’. This terminology extrapolates out to third-order synonyms and beyond.

In our experiment, we examine up to fourth-order synonyms, in order to observe the effects of far-reaching similarities. The synonyms are gathered into an ordered list in the following pattern: first the base word, then the first-order synonyms in the order given from the PyDictionary lookup, then the second-order synonyms of the first first-order synonym, in the order given from the PyDictionary lookup, then the second-order synonyms of the second first-order synonym, in the order given from the PyDictionary lookup, and so on for the remaining second-order synonyms, then the third-order synonyms of the first second-order synonym, in the order given from the PyDictionary lookup, and so on for the remaining third-order synonyms, and so on for any higher-order synonyms. The result creates an ordering in which all synonyms of order n are later in the list than all synonyms of order $n - 1$ but earlier in the list than all synonyms of order $n + 1$. Furthermore, within that ordering the earlier of an index an order n word possesses within the order n cluster the earlier an index all words derived from it will be found in the order $n + 1$ cluster. In addition, words from a PyDictionary lookup that match the base word or have already been found from earlier PyDictionary lookups are not added to the list again, ensuring that every word in the ordered list is unique.

This process is applied for both the base word and the word compared against it, generating two lists of synonyms extending out to fourth-order synonyms, and then the two ordered lists of synonyms are compared against each other for matching words. Each match found can be described by the indices of the word in each of the lists, which can be stored as a pair-value of integers for later use. The set of all pair-values so extracted from the ordered lists of words describes the similarity of the two words being compared.

When evaluating the pair values in order to process them into a coherent number, we have two main desired traits. The first desired trait is that lower index values represent higher similarity, as it suggests that the concepts behind the two base words matched within a relatively small conceptual distance, and the second desired trait is that more pair-values being present at once represents higher similarity, as it suggests that the concepts are close enough to share many different synonyms with each other. The equation that combines the set of pair-values into a single granular number is thus as follows:

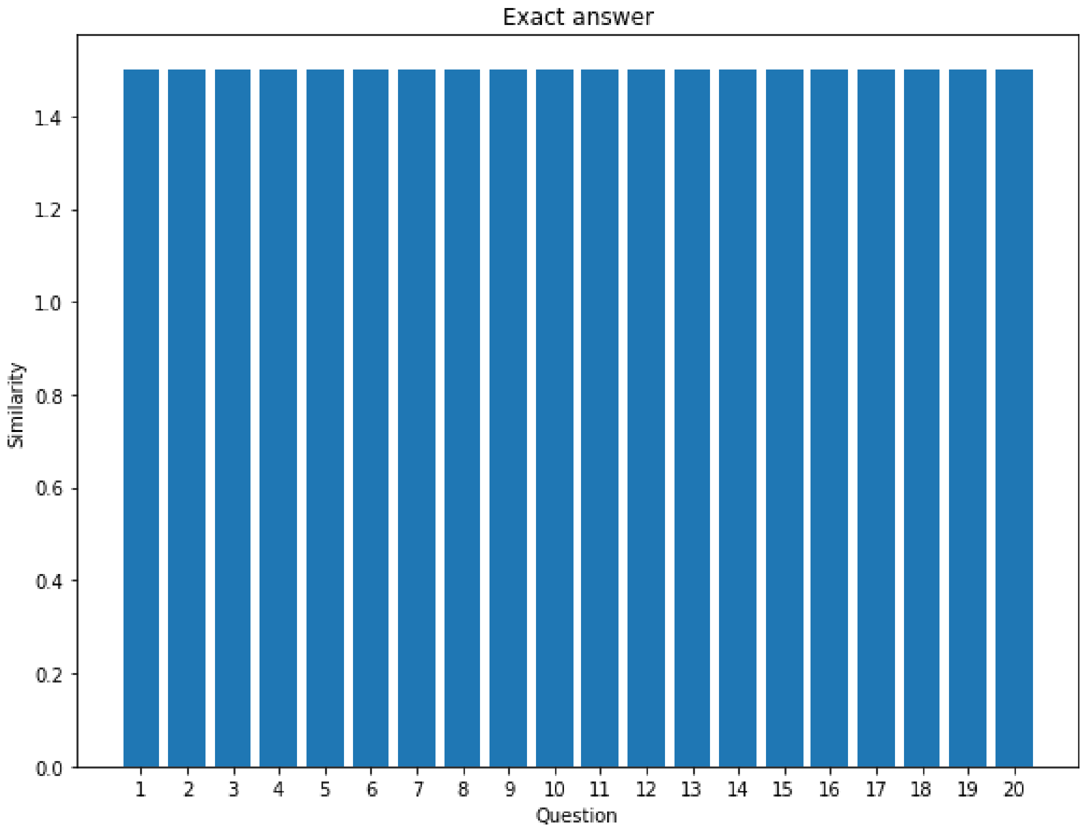
$$score = \sum_{i=1}^{n} \frac{1}{a_i + r_i} \tag{5.1}$$

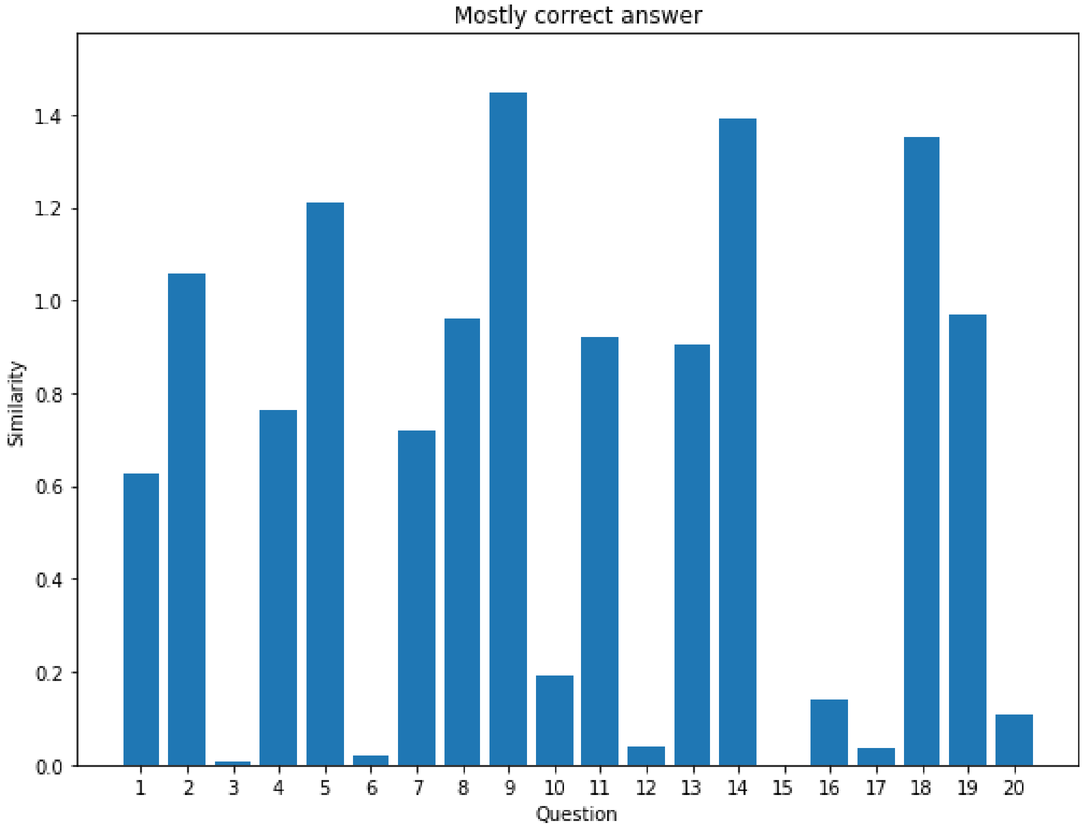
in which n represents the number of pair values found, a_i represents the first value of the ith pair, and r_i represents the second value of the ith pair. If the value produced is greater than 1.5 it is capped to that value, both to ensure exact matches produce the same similarity score and to reduce the difference between the 'exact match' category and the other three categories for a smoother gradient of comparison.

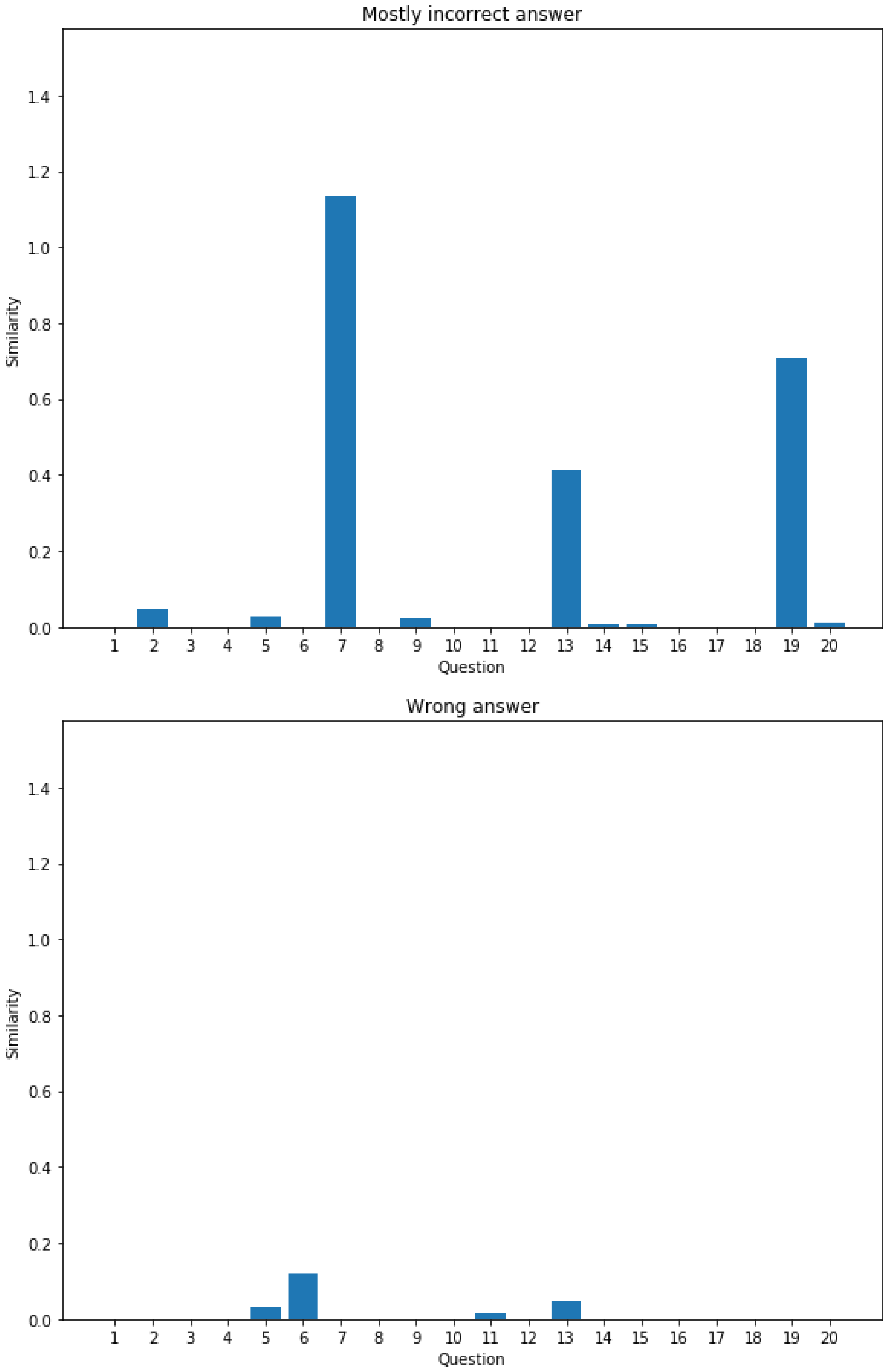
This comparison was tested with each of the base words in the dataset against each of the three words derived from that base word, as well as against itself as a baseline, to create four sets of 20 data points, each dataset representing the similarity score attained from using this process on the two words in question. The results are organized by the category of the second word and shown below in Figure 5.2:

In addition, the mean of each category has been gathered and placed directly side-by-side in Figure 5.3, so as to describe the average expected similarity score based on how close a match the input word is as compared to the expected word.

To further analyse the results of this experiment, three separate ROC curves were constructed, shown in Figure 5.4. The first ROC curve, 'Top 25% ROC Curve' assigned entries of the data class 'Exact answer' as 1 and all other entries as 0. The second ROC curve, 'Top 50% ROC Curve', assigned entries of the data classes 'Exact answer' and 'Mostly correct answer' as 1 and all other entries as 0, essentially dividing the dataset into an upper and lower half. The third ROC curve, 'Top 75% ROC Curve', assigned all entries of data classes other than 'Incorrect answer' as 1 and entries of the data class 'Incorrect answer' as 0.


Exact answer
Similarity
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Question


Mostly correct answer
Similarity
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Question



Mostly incorrect answer

Wrong answer

Figure 5.2: Results of Word Matching implementation

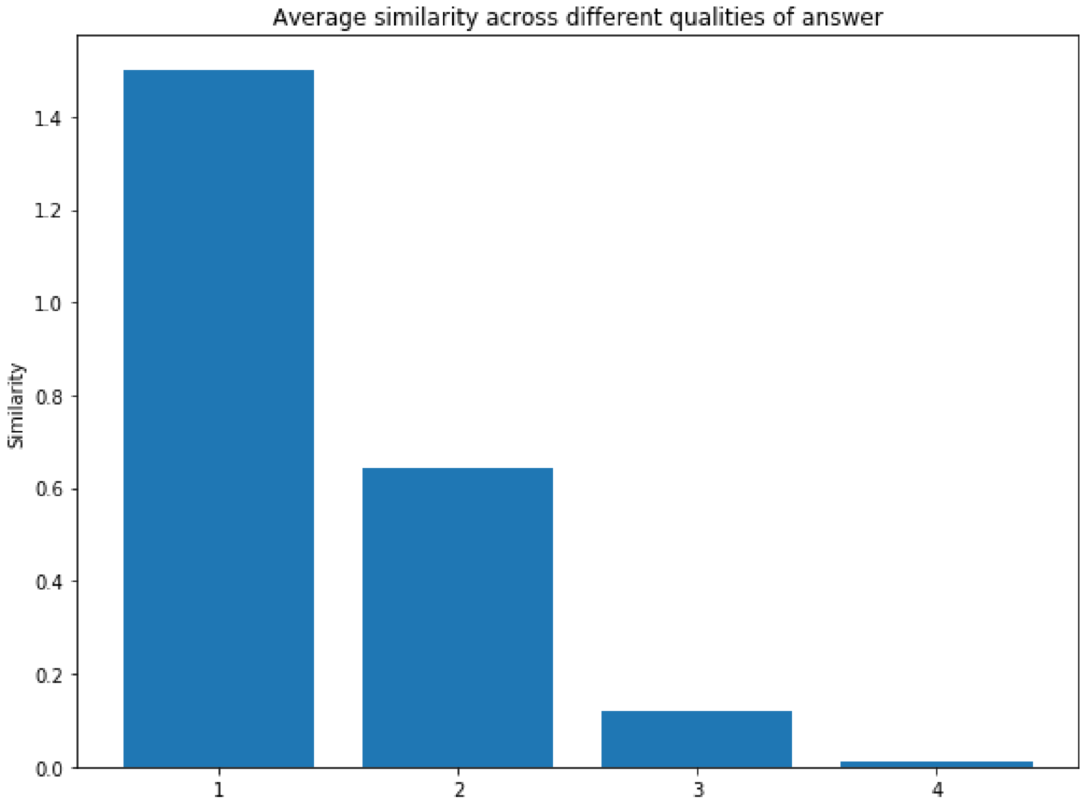


Figure 5.3: Average Word Matching Similarity

’Top 25% ROC Curve’ is a flat line with an area under curve of 1.00 due to the fact that all ’Exact answer’ entries have a higher value than all other entries. ’Top 50% ROC Curve’ is a smoother curve with an area under curve of 0.95, indicating that the entries in the data class ’Mostly correct answer’ are not as strictly superior to the rest of the dataset as the ’Exact answer’ entries were. ’Top 75% ROC Curve’ is less smooth than ’Top 50% ROC Curve’ and bears a lower area under curve of 0.85, indicating that there is greater overlap between the lowest data class of ’Incorrect answer’ and the data classes above it than there is overlap between the top half of the dataset, ’Exact answer’ and ’Mostly correct answer’, and the lower half of the dataset, ’Mostly incorrect answer’ and ’Incorrect answer’. Despite this, however, there still appears to be a distinguishable degree of separation, indicating that the ’Incorrect

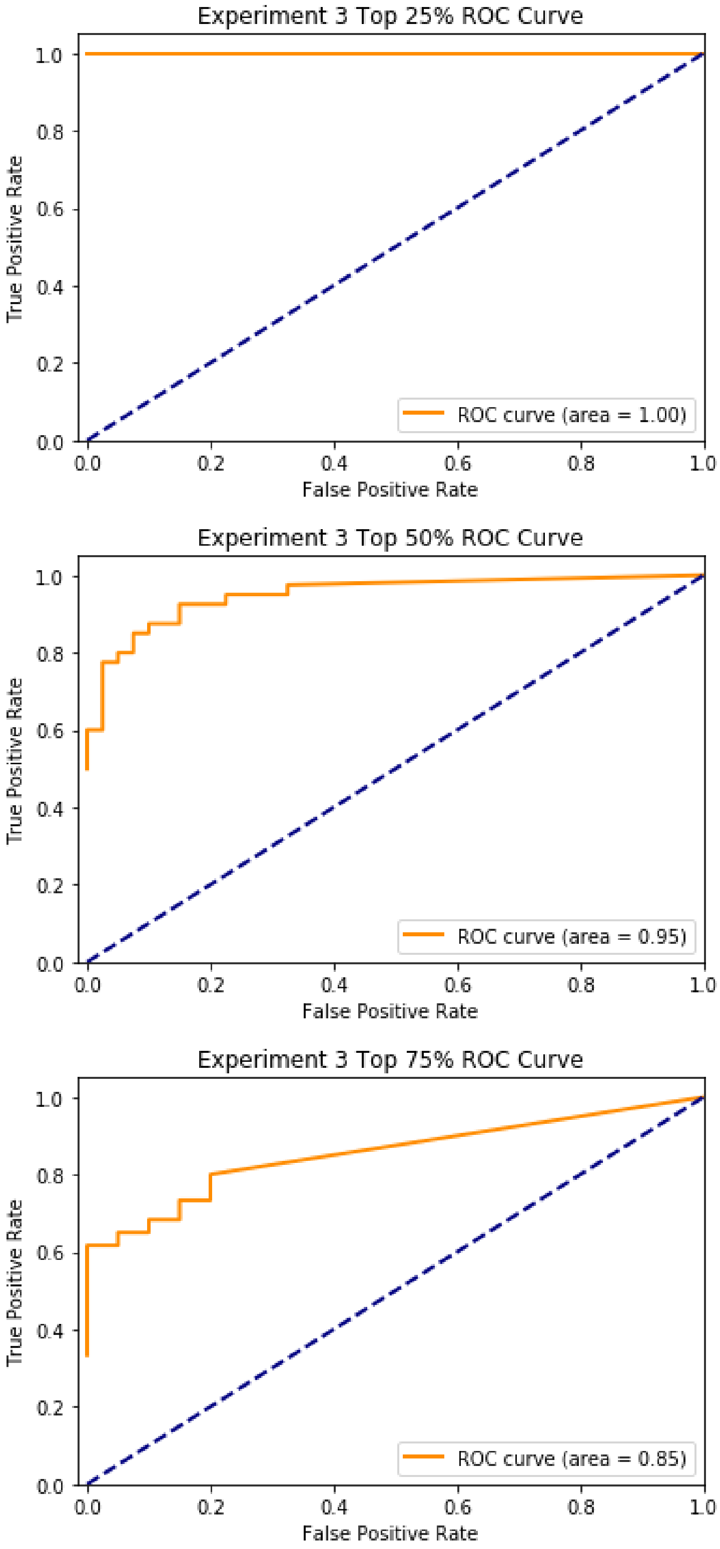


Figure 5.4: Experiment 3 ROC curves

answer' data class is still coherently different from the rest of the classes.

5.4 Discussion

The experiments conducted in this section each relate to integral elements of the final design that must be solved for AIAC to have viability as a system. The purpose of creating these experiments is to indicate the likelihood that everything proposed in this paper can be genuinely implemented into a real system.

The core difficulty in creating these experiments is the lack of real-world data to work with, as AIAC is designed to generate its own data during operation. Accordingly, without a fully implemented AIAC, we cannot access the data that it will eventually be trained on, nor gauge its performance on such data. Therefore, since the viability of AIAC needs to be demonstrated before it can be implemented, we manually generated custom data with an intent on being as authentic as possible to how AIAC is expected to generate data. This allows us to train the neural networks involved in the AIAC viability experiments to a standard as closely approximating AIAC's true operation as can be accomplished.

The first experiment investigates AIAC's ability to rank anomalies based on how useful they are likely to be for distinguishing between authentic and fraudulent user, and it achieves a total accuracy of 70.8% at that task. This, while not an extremely high accuracy, shows the presence of the core capability of the system to learn how to rank anomalies, accomplished with a comparatively small amount of manual input data. In addition, the ROC curve generated from the neural network's predictions indicate that the data corresponding to authentic and fraudulent anomalies are decently separated. The performance in an actual implementation is expected to be significantly higher, as the system is provided with significantly greater quantities of data drawn from actual user behaviour.

The second experiment investigates AIAC's ability to create natural language questions by choosing authentication challenge templates based on how likely they are to be comprehensible to the user, and it achieves an accuracy of 84.0% at that task. This score similarly shows the presence of the core capability of the system to learn how to create comprehensible authentication challenges, also accomplished with a comparatively small amount of manual input data. The performance in actual implementation is expected to be significantly higher, as the system is provided with significantly greater quantities of data drawn from actual user behaviour.

The final experiment investigates AIAC's ability to compare textual user responses to the expected answer with a level of granularity above and beyond simply checking for an exact word match, and the results shown in Figure 5.3 and Figure 5.4 indicate that such granularity can be achieved in the majority of cases. Higher performance can be attained by extending the settings to use higher-order synonyms, at the expense of more computational power, but even the current settings suggest a satisfactory ability to derive similarity metrics from two separate words.

In total, each of these experiments indicates that the fundamental principles of their operation are likely viable, which is promising for the overall viability of AIAC. These experiments are limited by a fundamental constraint stemming from the fact that with no fully-implemented AIAC all of the input data is manually generated and the principle of AIAC improving based on its own operation can only be approximated, not directly tested. Accordingly, the goal of these experiments is to determine what can be determined before implementation, specifically whether or not the underlying principles of AIAC's operation are sound. Under this goal, we believe that each experiment has been a success within the necessary constraints of this situation.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

Many modern authentication mechanisms make trade-offs between security and accessibility, with the necessary precautions or hardware needed to increase security limiting where the authentication can be applied and who can use it. As a result, there is a continual need for new authentication mechanisms that push the envelope on maximizing both security and accessibility so that we may improve on the systems created in the past and make authentication secure and convenient for everyone.

At the same time, recent developments in computing power and data gathering have made more Big Data applications possible, relying on very large amounts of data to forge new accomplishments in many different fields. The field of authentication is no exception, with new possibilities opening up as more information about users in larger quantities becomes available.

There exists a new authentication framework known as Just-in-time human dynamics based authentication engine (JitHDA). JitHDA is an authentication mechanism that leverages Big

Data to observe user behaviour, isolate anomalous events, and construct authentication queries based on these events. JitHDA operates on the expectation that the anomalous events it identifies represent memorable events in the user's life, thus ensuring that said user will have an easy time answering a question about the event in the time period shortly thereafter.

JitHDA can be subdivided into a small set of component softwares that fulfil certain tasks and connect with each other to compose the whole functionality. One of the main components of JitHDA, the collection of user data and processing of it into lists of anomalous events, has already been completed. The primary goal of this research is to produce a software design that is capable of implementing the remaining functionality needed to actualize JitHDA. This includes the process of turning anomalous events into authentication queries and the algorithm for authenticating the user based on the results of the query.

To achieve this, we designed a software which we call Autonomous Inquiry-based Authentication Chatbot (AIAC), using chatbot and machine learning technologies to solve the challenges of this process.

Given the structure of the data we were provided, in which anomalous events in the user's transaction history have already been recorded, we designed in full a mechanism by which that data can be used to create coherent authentication challenges with an already known answer so as to contrast the user's response against the known answer. The process by which AIAC does this is self-improving with the goal of maximizing the ability to discern between authentic and fraudulent users. We designed AIAC to compare user responses with the known answer in a robust and meaningful capacity. Even when the topic of the answer is not an easily comparable value such as a quantity of money, our design for AIAC is expected to be able to create granular and accurate comparisons to determine how correct the answer is. Lastly, AIAC is also self-improving with respect to its ability to create coherent authentication challenges, using user feedback in order to help determine how to structure a given authentication challenge so as to be maximally comprehensible. In aggregate, these components assembled according to our design are wholly sufficient for creating a fully-functioning authentication system.

AIAC's design is as accessible as passwords, requiring only text input from the user, while remaining secure by relying on recent life events, information only the user should know. AIAC also avoid some of the failure cases of passwords by automatically discarding old queries as time passes and the user performs more activities. In addition, the nature of AIAC avoids the security risk of requiring the user explicitly remember arbitrary information, which can cause them to externally record it in a way that allows other people to see it, closing off another security risk that passwords struggle with.

Other cutting-edge chatbot applications focus on gathering data from the user to narrow down a conclusion from a large possibility space, such as determining what emotion the user is feeling or what information they seek. While AIAC can also be said to be narrowing down a conclusion based on data provided by the user, AIAC clearly distinguishes itself in having already determined the answer to the questions it asks, and only deriving information from whether the user's answer corresponds to the expected answer, a quality not seen in other novel chatbot applications.

Meanwhile, recent natural language understanding applications focus primarily on the problems of sentiment analysis and multi-intent prediction. Among these works, AIAC's strategy of determining word similarity by comparing expanded lists of synonyms stands out as a novel approach in the field.

In order to demonstrate that a program like AIAC can be expected to function as described, we isolated the three most novel features of AIAC and created experiments to test their functionality:

The first experiment tested AIAC's ability to learn how to discern between useful and useless anomalies. To accomplish this, we manually simulated the results of authentic and fraudulent people answering different anomalies by allowing authentically-answered anomalies to view the expected answer of the anomaly while denying that information for fraudulently-answered anomalies.

The second experiment tested AIAC's ability to learn how to select which authentication challenge template to assign the chosen anomaly to. To accomplish this, we created a batch of authentication challenge templates and then manually rated the comprehensibility of each anomaly when applied to each template.

The third experiment tested AIAC's ability to compare keywords against each other in order to create a granular similarity score between the two. To accomplish this, we created a program that recursively found synonyms of input words and used it to determine the achieved similarity score between various pairs of words.

In the end, all of these experiments were successful, indicating the viability of the mechanisms to the extent that they can be indicated without a fully assembled AIAC. With this we feel confident saying that AIAC as currently designed is likely to be capable of fulfilling the needs of JitHDA and creating a new mode of authentication.

We expect that should AIAC be implemented as designed here, it would only need training data before it is ready for commercial application. Once sufficient training data in the appropriate format for the commercial application is found, AIAC should become fully operational and ready to use as a self-improving authentication chatbot.

Based on this we believe that organizations focusing on topics like communication or finance will be best suited for adoption of AIAC, given that they would both have a strong interest in data security and have the means to gather suitable amounts of data on their users to actualize the full potential of AIAC. This includes communications organizations such as Rogers or Bell Canada as well as banking institutions such as the Royal Bank of Canada or the Bank of Nova Scotia. We also believe that healthcare organizations would likely be well-suited to adoption of AIAC, for similar reasons of having a strong interest in data security as well as the opportunity to gather significant amounts of data on a per-user basis.

6.2 Future Works

While the design for AIAC has been laid out, and preliminary tests have been created to demonstrate the viability of its mechanisms, the system is meant to operate as a cohesive unit and thus it is impossible to get a truly accurate test of its capabilities without instantiating AIAC in full. The clear next step for the fulfilment of JitHDA is for a prototype of AIAC to be created according to the design presented here and for that prototype to then be tested with full authentication sessions in order to gather data on AIAC's true functionality.

In addition to performance metrics, the creation of a full instantiation of AIAC will also present the opportunity to fine-tune aspects of its design. Certain elements such as the exact nature of how incomprehension is to be signalled, whether it should be derived solely from the content of the user's response or if a separate button should be placed to directly signal it, can be examined in detail in the creation of the system to optimize the effectiveness of AIAC at its desired tasks.

This thesis lays out the design for a new system that will help create a new form of authentication only possible now in the age of Big Data. Once implemented, JitHDA will seek to prove itself as a useful alternative or supplement to password-based authentication systems by improving on their level of security while remaining just as accessible.

Bibliography

[1] I. I. M. Abu Sulayman and A. Ouda. User modeling via anomaly detection techniques for user authentication. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 0169–0176, 2019.

[2] Yosef Ashibani and Qusay H. Mahmoud. A multi-feature user authentication model based on mobile app interactions. *IEEE Access*, 8:96322–96339, 2020.

[3] D. Biswas. Privacy preserving chatbot conversations. In *2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 179–182, 2020.

[4] E. Cho, H. Xie, J. P. Lalor, V. Kumar, and W. M. Campbell. Efficient semi-supervised learning for natural language understanding by optimizing diversity. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1077–1084, 2019.

[5] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra. Visual dialog. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1080–1089, July 2017.

[6] L. Dostálek. Multi-factor authentication modeling. In *2019 9th International Conference on Advanced Computer Information Technologies (ACIT)*, pages 443–446, 2019.

[7] Y. Gahi, M. Lamrani, A. Zoglat, M. Guennoun, B. Kapralos, and K. El-Khatib. Biometric identification system based on electrocardiogram data. In *2008 New Technologies, Mobility and Security*, pages 1–5, 2008.

[8] K. J. Jose and K. S. Lakshmi. Joint slot filling and intent prediction for natural language understanding in frames dataset. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 179–181, July 2018.

[9] C. Kao, C. Chen, and Y. Tsai. Model of multi-turn dialogue in emotional chatbot. In *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 1–5, 2019.

[10] S. Kim and S. Kim. General authentication scheme in user-centric idm. In *2016 18th International Conference on Advanced Communication Technology (ICACT)*, pages 737–740, 2016.

[11] R. Kulkarni, H. Kulkarni, K. Balar, and P. Krishna. Cognitive natural language search using calibrated quantum mesh. In *2018 IEEE 17th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pages 174–178, July 2018.

[12] Adam Lally, Sugato Bagchi, Michael A. Barborak, David W. Buchanan, Jennifer Chu-Carroll, David A. Ferrucci, Michael R. Glass, Aditya Kalyanpur, Erik T. Mueller, J. William Murdock, Siddharth Patwardhan, and John M. Prager. Watsonpaths: Scenario-

based question answering and inference over unstructured information. *AI Magazine*, 38(2):59–76, Jul. 2017.

[13] Y. Lan, S. Wang, and J. Jiang. Knowledge base question answering with a matching-aggregation model and question-specific contextual relations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(10):1629–1638, Oct 2019.

[14] B. Liu, Z. Xu, C. Sun, B. Wang, X. Wang, D. F. Wong, and M. Zhang. Content-oriented user modeling for personalized response ranking in chatbots. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):122–133, 2018.

[15] Saci Medileh, Laouid Abdelkader, El Moatez Billah Nagoudi, Reinhardt Euler, Ahcène Bounceur, Mohammad Hammoudeh, Muath Alshaikh, Amna Eleyan, and Osama Khashan. A flexible encryption technique for the internet of things environment. *Ad Hoc Networks*, 106:102240, 06 2020.

[16] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, and Nader Kesserwan. Data-driven security for smart city systems: Carving a trail. *IEEE Access*, 8:147211–147230, 2020.

[17] D. Mujtaba and N. Mahapatra. Recent trends in natural language understanding for procedural knowledge. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 420–424, 2019.

[18] Alma Oracevic, Selma Dilek, and Suat Ozdemir. Security in internet of things: A survey. In *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6, 2017.

[19] A. Ouda. A framework for next generation user authentication. In *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–4, 2016.

[20] F. Patel, R. Thakore, I. Nandwani, and S. K. Bharti. Combating depression in students using an intelligent chatbot: A cognitive behavioral therapy. In *2019 IEEE 16th India Council International Conference (INDICON)*, pages 1–4, 2019.

[21] V. A. Prasad and R. Ranjith. Intelligent chatbot for lab security and automation. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–4, 2020.

[22] B. Rychalska, H. Glabska, and A. Wroblewska. Multi-intent hierarchical natural language understanding for chatbots. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 256–259, Oct 2018.

[23] B. Setiaji and F. W. Wibowo. Chatbot using a knowledge in database: Human-to-machine conversation modeling. In *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pages 72–77, Jan 2016.

[24] P. Srivastava and N. Singh. Automatized medical chatbot (medibot). In *2020 International Conference on Power Electronics IoT Applications in Renewable Energy and its Control (PARC)*, pages 351–354, 2020.

[25] I. I. M. Abu Sulayman and A. Ouda. Human trait analysis via machine learning techniques for user authentication. October 2020.

[26] N. T. Thomas. An e-business chatbot using aiml and lsa. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2740–2742, Sep. 2016.

[27] Tayfun Tuna, Esra Akbas, Ahmet Aksoy, M Abdullah Canbaz, Umit Karabiyik, Bilal Gonen, and Ramazan Aygun. User characterization for online social networks. *Social Network Analysis and Mining*, 6:104, 11 2016.

[28] P. Voege and A. Ouda. A study on natural language chatbot-based authentication systems. In *2021 International Symposium on Networks, Computers and Communications (ISNCC): Trust, Security and Privacy*, October 2021.

[29] P. Voege, I. I. M. Abu Sulayman, and A. Ouda. Smart chatbot for user authentication. *Information Processing & Management*, 2022.

[30] T. Zhu, Z. Qu, H. Xu, J. Zhang, Z. Shao, Y. Chen, S. Prabhakar, and J. Yang. Riskcog: Unobtrusive real-time user authentication on mobile devices in the wild. *IEEE Transactions on Mobile Computing*, 19(2):466–483, 2020.

Curriculum Vitae

Name:	Peter Voege
Post-Secondary Education and Degrees:	Western University London, ON 2015 - 2019 B.SE.
Related Work Experience:	Teaching Assistant Western University 2019 - 2021

Publications:

(1) P. Voege and A. Ouda. A Study on Natural Language Chatbot-based Authentication Systems. In 2021 International Symposium on Networks, Computers and Communications (ISNCC): Trust, Security and Privacy, 2021.

(2) P. Voege, I. I. M. A. Sulayman, and A. Ouda. Smart Chatbot for User Authentication. In Information Processing & Management, Special Issue on Leveraging Text and Social Analytics for Business Intelligence, 2022, under review.