electronics

MDPI

Article

Smart Chatbot for User Authentication

Peter Voege *, Iman I. M. Abu Sulayman and Abdelkader Ouda

Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada
* Correspondence: pvoege2@uwo.ca

Abstract: Despite being the most widely used authentication mechanism, password-based authentication is not very secure, being easily guessed or brute-forced. To address this, many systems which especially value security adopt Multi-Factor Authentication (MFA), in which multiple different authentication mechanisms are used concurrently. JitHDA (Just-in-time human dynamics based authentication engine) is a new authentication mechanism which can add another option to MFA capabilities. JitHDA observes human behaviour and human dynamics to gather up to date information on the user from which authentication questions can be dynamically generated. This paper proposes a system that implements JitHDA, which we call Autonomous Inquiry-based Authentication Chatbot (AIAC). AIAC uses anomalous events gathered from a user's recent activity to create personalized questions for the user to answer, and is designed to improve its own capabilities over time using neural networks trained on data gathered during authentication sessions. Due to using the user's recent activity, they will be easy for the authentic user to answer and hard for a fraudulent user to guess, and as the user's recent history updates between authentication sessions new questions will be dynamically generated to replace old ones. We intend to show in this paper that AIAC is a viable implementation of JitHDA.

Keywords: machine learning; authentication; natural language understanding; big data; chatbots

Citation: Voege, P.; Abu Sulayman, I.I.M.; Ouda, A. Smart Chatbot for User Authentication. *Electronics* **2022**, *11*, 4016. https://doi.org/10.3390/electronics11234016

Academic Editor: Zbigniew Kotulski

Received: 1 November 2022
Accepted: 26 November 2022
Published: 3 December 2022

1. Introduction

User authentication is a vital component of daily operation of the modern world. Many applications and institutions rely on complete security for their users' accounts, and in a digital environment where an access request can come from any location at any time regardless of authenticity, it is imperative to have robust authentication technology.

Authentication techniques can be divided into four different categories. You can authenticate someone based on something they know, something they have, something they are, and something they do. Authentication based on something you know takes the shape of systems such as passwords and PIN numbers. Authentication based on something you have takes the shape of systems such as keys or ID cards. Authentication based on something you are takes the shape of biometrics such as fingerprint or iris scans. Authentication based on something you do takes the shape of systems such as analysing a user's typing habits.

'Something you know' authentication is very convenient and accessible, as the authentication merely requires inputting the information you have stored into your head, which can be completed in seconds on almost any authentication medium. However, it is also very insecure, as it is easy for the information to leak and when that happens any bad actor with the information is able to impersonate you. 'Something you have' authentication is only moderately convenient and accessible, as you must have the object in question available to you if you want to perform authentication, but is somewhat more secure than 'something you know' authentication because there is only one copy of the object to misplace and it cannot be taken without your notice. 'Something you are' authentication is highly secure, as it is difficult to fake biometric scans, but is comparatively inconvenient and inaccessible

as you generally need specialized hardware present to perform the biometric scan. 'Something you do' authentication is a new form of authentication that operates by using user behavioural patterns, which can be very difficult to fake [1]. Typing rhythm and voice patterns are examples of the kind of user behaviour that qualifies for 'something you do' authentication, as they are unique to each user and difficult to impersonate [2].

None of these techniques are perfect. There are a great many considerations that a successful authentication system must perform well on [3], and in any given context we must choose one based on its situational advantages and disadvantages as there is no universally ideal authentication system. The most common method by far is 'something you know' authentication, which faces a continual struggle to maintain strong security without compromising accessibility [4]. To accomplish this, we propose a synthesis of 'something you do' and 'something you know' authentication achieved by analysing a user's behaviour in their life activities and then questioning them on said activities. This behaviour-driven authentication system applies the security of 'something you do' authentication to the accessibility of 'something you know' authentication to create a strong, convenient method of authentication that anyone can make use of.

This system, which we call Autonomous Inquiry-based Authentication Chatbot (AIAC), will be a chatbot interacting with the user which continuously generates new questions based off of recent data, such that information used for authentication is quickly rendered useless for exploitation by bad actors, even if they do learn what it is. As a result, relying on 'something you know' authentication will not carry the risk that bad actors will acquire the user's authentication information and impersonate them, thus making it much more secure as an authentication strategy.

Ouda et al. [5] has developed a new authentication framework based on "something you do" authentication principles and leveraging Big Data analytics to create dynamic and personalized authentication challenges. This framework is made up of three core components, as shown in Figure 1, each developed separately from different perspectives.

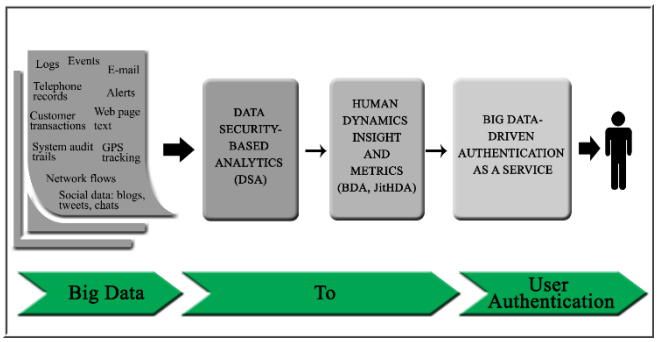

Figure 1. The components of Dr. Ouda's authentication framework [5].

The first component, known as Data Security-based Analystics (DSA), describes ways to leverage Big Data analytics to form accurate up-to-date models of user behaviour. The second component, known as Big Data-driven Authentication tool (BDA), identifies patterns in DSA models and then uses said patterns to detect anomalous activity from the user and assemble a security profile about the user. The third component, known as Just-in-time Human Dynamics based Authentication engine (JitHDA), uses the user profiles to dynamically create secure authentication questions in real-time which derive from the user's recent behaviour [6].

When all of these components are used in sequence, the complete authentication framework is formed, creating a novel 'something you do'-based method of authentication.

The goal of this paper is to present a module that achieves that synthesis for the purposes of implementing JitHDA [5]. JitHDA is an authentication chatbot that focuses on forming authentication challenges just-in-time in order to remove the increasing temporal risk of static authentication information and on constructing the authentication challenges

from observed human dynamics. JitHDA is exactly the synthesis of 'something you do' and 'something you know' authentication that we seek to make possible.

A fully implemented JitHDA would have the following qualities: (i) The expected answer should be based on a recent action that the user has taken. (ii) The question created should be randomly chosen from a list of possible questions. (iii) Questions should not be repeated. The complete authentication framework is composed of two modules: the location of anomalous events from recent user actions, and the construction of authentication challenges from the located anomalous events. JitHDA describes the latter module, and AIAC is designed to implement JitHDA.

Our novel advancement towards this goal is a system that uses machine learning to select optimal anomalous events from a user's recent history for the purposes of creating useful authentication challenges, and then further using machine learning to optimize the construction of natural language questions out of the selected anomalous event. AIAC will make use of both of these qualities to create smart authentication sessions calibrated to efficiently and carefully sort authentic users from fraudulent users based purely on information that an authentic user would know.

The rest of this paper will be structured as follows: Section 2 will cover related work to this subject, Section 3 will cover the problem we seek to address, Section 4 will cover our solution to the problem, and Section 5 will cover the experiments we performed to establish the validity of the proposed solution.

2. Related Work

To understand the relevance of AIAC, we must look at the other works in this domain to see how our contribution compares to what they have accomplished, and what overlap in purpose can be found.

B. Liu et al. [7] 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. This is relevant to JitHDA's goal of personalized chatbot sessions with the user, but approaches that concept by dynamically integrating user posts in real-time instead of creating personalized profiles of the user to construct sessions from.

C. Kao et al. [8] 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 paper is also relevant to JitHDA's goal of personalized chatbot sessions with the user, but accomplishes it from real-time use of user input instead of pre-assembled user information.

F. Patel et al. [9] 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 too is relevant to JitHDA's goal of personalized chatbot sessions with the user, but this paper also achieves its goal by examining specific user statements in real-time instead of constructing the personalized session information from knowledge acquired beforehand.

P. Srivastava et al. [10] describes a chatbot which can automatically diagnose a person's medical troubles. It begins from a position of ignorance and converses with the user until it can determine a shortlist of possible illnesses and eventually a most probable illness, and then proceeds to make recommendations for dealing with the problem. The system is similar to AIAC in that it iteratively asks dynamically chosen questions of a user in an attempt to form a model of the user. This paper is relevant to JitHDA's goal of asking questions of the user and refining a model from the responses until a conclusion can be

reached. Its main difference is that it starts from a position of ignorance and refines its questions as it advances to prune a wide possibility space to a single option, whereas JitHDA starts with ample information and uses it to create questions that discern between two possible states. The strategies and goals are similar, but ultimately approach the task in significantly different ways.

T. Zhu et al. [11] deals with authentication of mobile devices with the intent of warding against the threat of someone's phone getting into the wrong hands. It discusses various biometric-based ('something you are') authentication and chooses to use widely used motion sensors found in many mobile devices to learn the patterns and behaviour of the authentic user for the purpose of authentication. This paper is relevant to JitHDA's goal to use human dynamics to augment authentication capabilities, but focuses more on the physical handling of a device rather than a user's knowledge of their past behaviour.

S. Kim et al. [12] 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 paper focuses on the tradeoff between security and convenience that JitHDA is designed to remedy, but otherwise does not help create a JitHDA process in and of itself.

L. Dostálek et al. [13] 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 is 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 paper is another angle on the tradeoff between security and convenience, as it allows a system to only enable the inconvenient security measures when there is reason to suspect that something is amiss. While it does not help implement JitHDA, a JitHDA implementation would likely be very useful in such a system.

T. Tuna et al. [14] 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.

While many of these papers provide novel and interesting advancements to chatbot and authentication technology, none of them are specifically applicable to implementing a JitHDA system. However, based on the favourable results of the experiments conducted above, we believe that AIAC is exactly suited to implementing JitHDA.

3. The Need for a Stronger Authentication

An ideal authentication system should be maximally (i) secure, (ii) accessible, and (iii) convenient. Such a system would allow any authentic user to authenticate themselves from any access point with minimal effort, while simultaneously rendering it impossible for any bad actors to fraudulently authenticate themselves.

When a situation does not permit achieving all of these goals at once, trade-offs must be made. 'Something you know' authentication often performs very well on ease of use, and is consequently a favoured option for many applications, but can be insecure if the bad actor acquires the relevant knowledge. 'Something you are' authentication can be extremely secure with fingerprint or iris scans that a bad actor would have no way to replicate [15], but can be very inaccessible if you need specialized hardware to perform the task adequately. Many recent applications combine 'something you know' authentication

with 'something you have' authentication with Two-Factor Authentication systems that rely on both knowledge of a password and possession of a specific phone in order to be authenticated. This configuration has higher security than pure 'something you know' authentication on account of both systems needing to fail at the same time and the two systems not having overlapping vulnerabilities, at a minor cost to convenience.

The most core problem with 'something you know' authentication is the risk that the information used for such systems, such as passwords or recovery questions, can eventually find their way into the hands of bad actors, whereupon they become able to easily pretend to be the user from then on. The high danger of this stems from the fact that passwords, recovery questions, and other similar methods tend to stay the same over long periods of time, giving a bad actor plenty of time to find a user's information. As an example, a recovery question based on the user's first pet will always have the same answer no matter when it is asked, and a bad actor therefore has all the time in the world to figure out the relevant information and apply it to breaking the security of the question.

As such, properly secure 'something you know' authentication requires three qualities. The first quality is that the information be hard to guess from a fraudulent user, the second quality is that the information be easy to recall from an authentic user, and the third quality is that the information not be repeated from previous authentication sessions. Passwords do not achieve this, instead being complex, repeated across many authentication sessions, and only hard to guess if the user actively chooses hard to guess passwords. A properly implemented JitHDA would be able to accomplish this, as JitHDA uses no repeat questions and builds each question from the recent actions of the user, which should be easy for authentic users to remember and hard for fraudulent users to guess. Therefore, a two-factor-authentication system utilizing both passwords and JitHDA will be much more secure than a one-factor-authentication employing only passwords.

4. The Proposed Autonomous Inquiry-Based Authentication Chatbot

The goal of this paper is to introduce an implementation of JitHDA that improves the security of 'something you know' authentication by dynamically changing the information required for authentication while retaining the high accessibility and convenience typical of the category. JitHDA as an authentication system is currently missing one core system: the user-facing element that generates novel questions for the user to answer and grants or bars access based on the user's answer. It is that system which we seek to implement.

Autonomous Inquiry-based Authentication Chatbot is a chatbot interacting with users in real-time when they seek to authenticate themselves which asks users questions about recent, dynamically chosen anomalous events in their life in order to determine their authenticity. By using questions derived from only recent information, it will become exceedingly difficult for a bad actor to acquire the information necessary to pass authentication before the questions become obsolete, whereas the authentic user will always be able to answer the questions from recent personal memory. With all the events that occur in a person's life, it is unlikely that even those close to the authentic user will know many of the relevant pieces of information.

We propose using a chatbot here because it allows for quick and accessible communication between the user and the authentication system in natural language. Being a chatbot rather than a single text input field as passwords prefer allows for repeated iterated questions in order to raise the confidence level of AIAC's estimation as high as it needs to. The process will be straightforward and intuitive from the user's perspective yet meaningfully informative for the authentication system.

During an authentication session, AIAC will be provided a user profile on the account in question, containing anomalous behaviour identified from user patterns and updated in real-time as new user data becomes available [16]. AIAC will automatically select anomalies out of the profile and construct out of them a question in natural language and a set of acceptable answers to the question, to present to the user as a smart authentication

challenge. It is expected that AIAC will be employed for platforms which have access to Big Data related to their users, collected through fair and ethical means.

AIAC will parse the user's response with Natural Language Understanding technology and compare the extracted answer to the set of acceptable answers to the question. If the user's response is very similar to the expected answer then it is likely that the user is authentic, and if the response is very dissimilar to the expected answer then it is likely that the user is fraudulent. AIAC will then repeat the process with a new question, iterating until the cumulative estimation of the user becomes sufficiently clear as either authentic or fraudulent, at which point the appropriate action can be taken.

A core functionality of AIAC will be its ability to refine its own operation by using the data that it collects in the process of its operation. The part of AIAC responsible for selecting anomalies to construct challenges from will update itself based on how well the chosen questions helped distinguish between authentic users and fraudulent users, so as to allow AIAC to gain an understanding of which questions authentic users will be able to correctly answer reliably but fraudulent users will not be able to correctly answer. In addition, the part of AIAC responsible for creating a challenge to the user in natural language will update itself based on whether or not the user understood the question, so as to allow AIAC to gain an understanding of how to describe a given anomaly as a coherent question that the user will understand.

The methodology of the proposed model is presented in the following subsections.

4.1. The Overall Process

The overall behaviour of AIAC is illustrated in Figure 2.

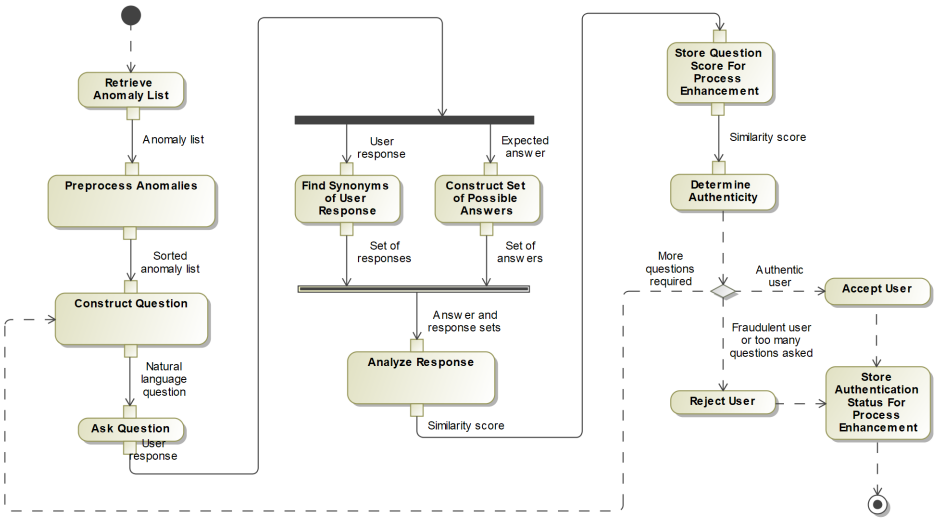

Figure 2. The overall process.

The primary input to AIAC is user profiles, automatically generated for each user. User profiles contain lists of anomalous behaviour collected from an anomaly detection algorithm which automatically determines whether a recorded event is normal or anomalous [16,17].

For our purposes, an anomalous event is something that deviates from the usual behaviour of the person in question. It is, however, difficult to meaningfully define the boundaries of anomalous vs. normal behaviour, and as such there are many different anomaly detection techniques attempting to solve the problem as efficiently as possible. Anomaly detection techniques are used in Big Data applications, applications related to user profiles, and user authentication systems.

The data used for AIAC's operation will be specific to each task AIAC is implemented for, since different institutions have access to different data and different demographics. The practice of using Big Data to enhance the user experience has been steadily becoming

more common recently, which suggests that over time the number of places where AIAC can be implemented will similarly rise. Nonetheless, it remains true that AIAC is limited to those systems that can adequately gather Big Data for use in AIAC.

As example of one of the datasets used for AIAC is the BankSim payments simulator. BankSim does agent-based simulations of bank payments based on aggregate data provided by a bank in Spain [18]. BankSim's main purpose is to generate useful synthetic data for fraud detection research. The data provided by BankSim contain no personal information or other private data, allowing it to be ethically used in academia. The synthetic data provided by BankSim contains 594,643 data values, of which 7200 are fraudulent transactions. A more detailed breakdown of its properties can be seen in Table 1.

Table 1. Overview of BankSim's data.

Data Set Name	Synthetic Data Set from a Financial Payment System
Data set features number	10
Data set observation number	594,643
Data set place	Spain
Normal-Anomalous percentage	98.79–1.21

The user profiles are created using DSA and BDA in combination to take raw input data of recorded user events and from that determine which events are normal and which events are anomalous. The software to create these models has already been made, and a variant customized for AIAC will be made to support its functionality.

The data gathered from the activity includes the user ID, the nature of the action, the amount associated with the action, and the timestamp. Beyond this, some further features are derived based on the anomaly detection process. An 'anomaly type' feature is created based on the feature in the original dataset that was responsible for it being determined to be anomalous. An 'unexpected observation' feature is created from the feature in the original dataset corresponding to the 'anomaly type' feature. Lastly, an 'expected behaviour' feature is created based on the user's overall statistics, to contrast against the 'unexpected observation' feature. The 'expected behaviour' feature is calculated as follows: (i) If the anomaly type is the time feature, the expected behaviour is the most frequent time recorded for that category for that user. (ii) If the anomaly type is the amount feature, the expected behaviour is the average amount recorded for that category for that user. (iii) If the anomaly type is the nature of the action, the expected behaviour is the most common category recorded for that user. (iv) If the anomaly type is the merchant ID, the expected behaviour is the most common merchant ID recorded for that user. The structure of an anomaly entry is shown in Figure 3.

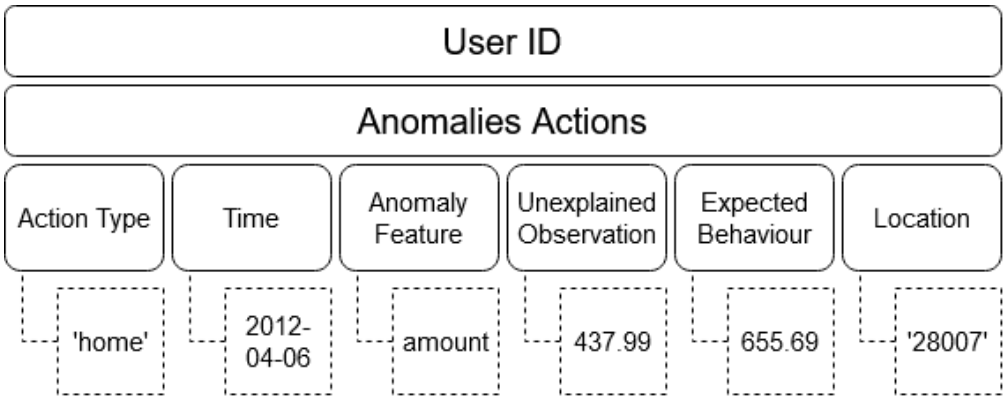

Figure 3. Anomaly structure.

As a result, the full structure of an anomaly is as follows: (i) The user ID, consistent across the profile (ii) The nature of the action recorded, or the type of transaction. (iii) The date the action occurred on. (iv) The nature of the anomalous action that led to it being recorded in the profile. (v) The specific value of the anomalous element of the action. (vi) The expected value for that element of the action. (vii) The location the action occurred at.

The profiles will hold a fixed number of these entries (n), such that the profile contains the n most recent anomalies recorded. The suite of questions AIAC will use in its authentication questions will be derived from the anomalies in the current profile and update as the profile updates.

Once presented with a new or updated user profile, the first step of AIAC's operation will be to select the anomalies most suitable to create questions out of. This will be accomplished by calculating a rating for each anomaly using a neural network and then using the ratings to sort the list from most suitable to least suitable.

The neural network will be feedforward, using labelled data for the purposes of supervised learning, connecting an input describing one anomaly to an output variable with 'deep layers' in-between to allow for the extraction of more abstract relationships and connections from the input data. Only some parts of an anomaly profile can be readily converted to inputs for the neural network, and the remaining parts will necessarily be excluded from consideration.

With foreknowledge of what inputs are expected, descriptions of an action type or anomaly type can be one-hot encoded with a boolean feature for each known possibility. Descriptions of time or money can be represented numerically. However, the User ID and zip codes are not useful for our purposes: the User ID does not contain any meaningful information about the anomalous event, and zip codes are likely to be highly unreliable in creating comprehensible information for authentication challenges. Accordingly, we exclude these features from the input of the neural network.

The output of the neural network is a single output node, giving an output number that will serve as the ranking for the anomaly.

A core functionality of AIAC is its ability to improve its functionality from the data it gathers during its operation. This neural network is the first half of this mechanism, where AIAC continually refines its ability to select anomalies that help distinguish between authentic and fraudulent users.

AIAC does this by keeping track of each anomaly throughout the authentication process, and recording the user's answer to the authentication challenge created by the anomaly. Once the authentication session ends and AIAC has determined whether the user is authentic or fraudulent, that result can be compared to the user's response to determine whether the question was helpful or detrimental. For instance, a question that a fraudulent user gets right is not helpful, and neither is a question that an authentic user gets wrong, but questions that fraudulent users get wrong and authentic users get right are useful. This comparison can be used to make labels for any given anomaly that describes how useful it was to AIAC's authentication process, and these labels can be used to train the neural network to select the most useful anomalies and ignore the least useful ones.

In the cases where an authentication session is terminated early, it will be impossible to determine for sure whether the user was authentic or fraudulent, and so anomalies sampled in that session will not be usable.

A diagram of the structure of such a neural network can be seen in Figure 4.

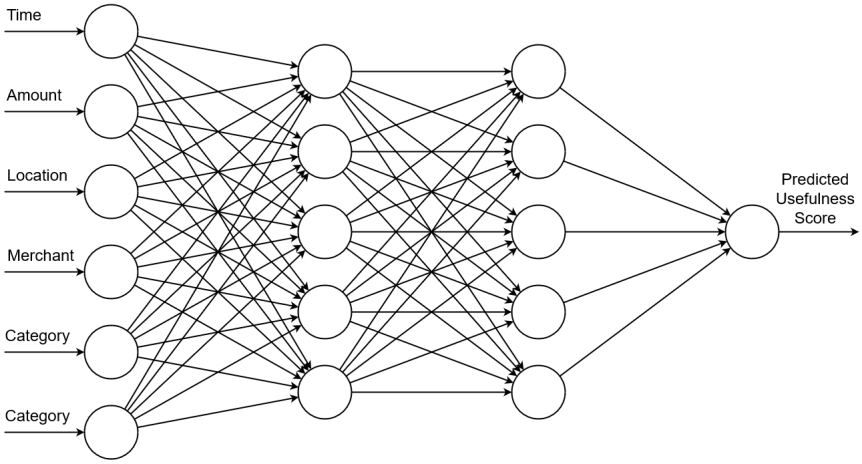

Figure 4. A feedforward neural network with one output.

4.2. Preprocessing the Anomaly Profile

Once the sorted list is created, the highest rated anomaly on the list will be used to create the first authentication challenge. This process is illustrated in Figure 5.

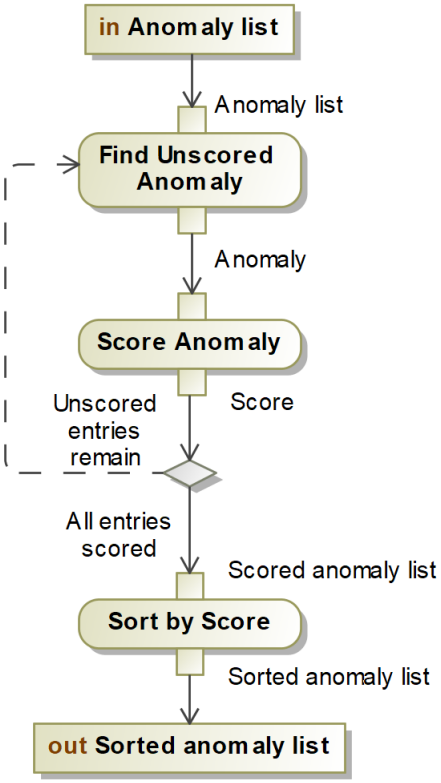

Figure 5. Preprocessing the anomaly profile.

The information within an anomaly can be used to create a question-answer pair, in which the user is queried with the constructed question and the user response is compared against the constructed answer. The user's behaviour, encoded in the anomaly and the constructed question, is knowledge that only the authentic user should know, and as a result the authentic user is the only person who should be able to produce the correct answer to the question.

In order to turn a given anomaly into a coherent question-answer pair, AIAC must identify the most suitable piece of information within the anomaly to use for the answer,

with the remaining pieces of information contained within the anomaly used to create a suitable question in natural language that queries the user for the information contained in the answer.

For example, suppose an anomaly is chosen with the following characteristics: (i) Happens on 15 July 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 "At what store did you purchase new shoes on 15 July?" with the expected answer being "Wal-Mart". 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.

4.3. Forming a Natural Language Question

It is difficult to dynamically generate natural language sentences that are reliably coherent and adequately convey the intended information, and so AIAC will circumvent the need to dynamically generate natural language by means of sentence templates. These sentence templates are pre-set sentences designed by humans to be coherent and meaningful queries, but which come with placeholder identifiers in the place of the keywords of the sentence. As such, AIAC can create a coherent natural language query by extracting information from the anomaly it wishes to query with and then insert the keywords into the chosen question template.

Figure 6 shows a sentence template represented conceptually, showing the breakdown of fixed text and specific keyword placeholders. These placeholders can be broken down according to English grammatical rules, in case flexibility is required.

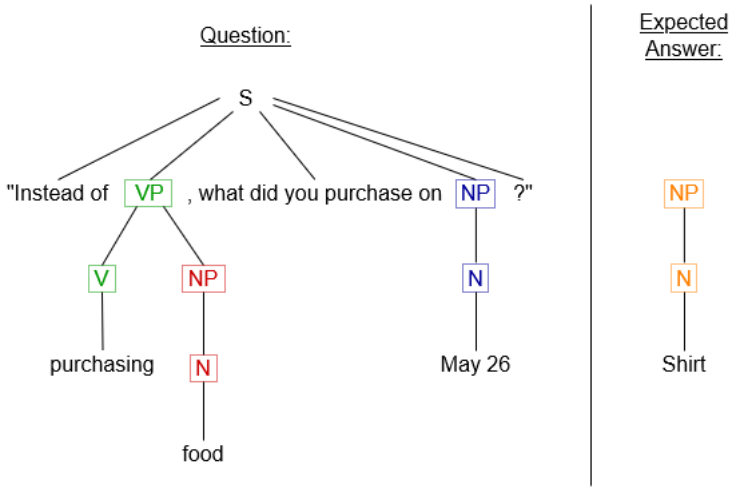

Figure 6. A full sentence template.

For example, we can take the anomaly described in Table 2 and fit it into a hypothetical authentication challenge template. With a question of "On <time>, how much money did you spend on <action>?" and an answer of "<amount>.", we can insert 'the 26 May into the <time> placeholder, 'health services' into the <action> placeholder, and '$646.86' into the <amount> placeholder. The text that is then presented to the user would be "On the 26th of May, how much money did you spend on health services?" with an expected answer of "$646.86".

Table 2. Details of BankSim's data.

User ID	Action	Time	Anomaly Type	Unexpected Observation	Expected Behaviour
'C1350963410'	'es health'	26 May 2018	amount	646.86	169.9580556

And important element for AIAC's configuration will be the protection of user privacy. It is imperative that, when the appropriate authentication challenge templates are created for AIAC's use case in a given implementation, they do not expose the user's private information to anyone who would be watching. Thus, authentication challenge templates should be designed such that the private information is only used in the answer section of the template, which the user cannot see.

There is a second choice that AIAC must now make, however, and that is which sentence template to use for the anomaly at hand. The same sentence structure and query format will not work for every anomaly that the query is being made of, and so there must be a variety of possible sentence templates to choose from. When it comes time to generate a question from an anomaly, AIAC will have to decide which question template is most likely to provide the user with a useful comprehensible authentication challenge that allows them to understand and answer the question coherently. To accomplish this, AIAC will need a second feedforward neural network to determine the most comprehensible sentence template for a given anomaly.

The input to this neural network will again be the details of the anomaly in question, formatted in the same arrangement as with the first neural network. However, instead of one output node, there will be an output node for every sentence template within the system. The neural network will assign each sentence template its own score based on suitability to deliver the anomaly, which we can then use to create a sorted list of the sentence templates.

While not sufficient in and of itself to determine the optimal authentication challenge template, the anomaly type is the feature that most directly correlates with suitability, enough that it can be possible to assist the neural network's operation by weighting the outcomes based on it. Authentication challenge templates that rely on a specific anomaly type and thus would not have enough information to create any sort of coherent authentication challenge can be assigned a weight of zero, while authentication challenge templates that expect the anomaly type that is currently present can be assigned a more favourable weight than average.

The other half of AIAC's ability to improve its functionality comes into play here. We wish to train the neural network to select templates based on maximizing comprehensibility of the presented question, which we can measure by whether the user appears to have understood the question or not. If the user indicates incomprehension, we can label that anomaly output pairing as unhelpful, and use that to train the neural network to pick better-suited templates in the future.

In addition, in cases where the authentication session terminates early, the questions they have answered are still valid targets for determining if the user comprehended the questions presented or not, and thus for creating labelled data.

A diagram of the structure of such a neural network can be seen in Figure 7.

Once the best template is chosen, AIAC extracts the required data from the anomaly and converts it into the structure required by the template, finishing the construction of the natural language question, which is then presented to the user. AIAC communicates with the user through a chatbot format, so the user will respond to the question by typing whatever response they think is correct. The full process of creating the question from a given anomaly is illustrated in Figure 8.

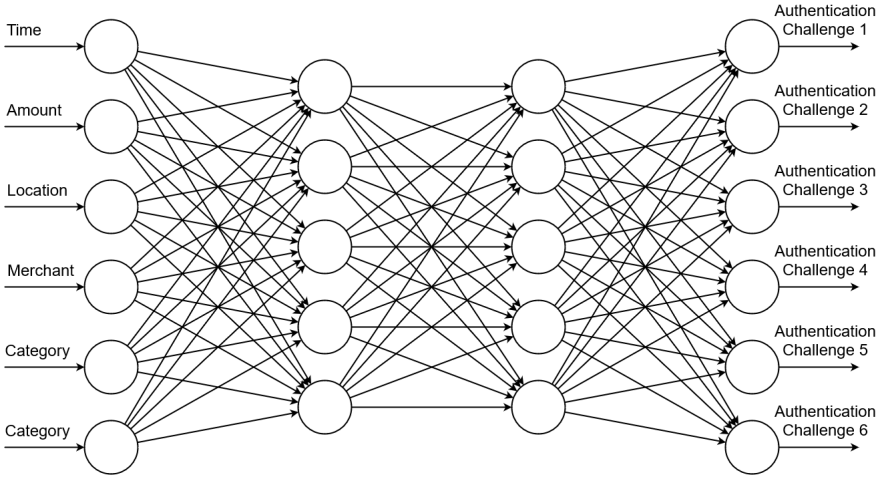

Figure 7. A feedforward neural network with many outputs.

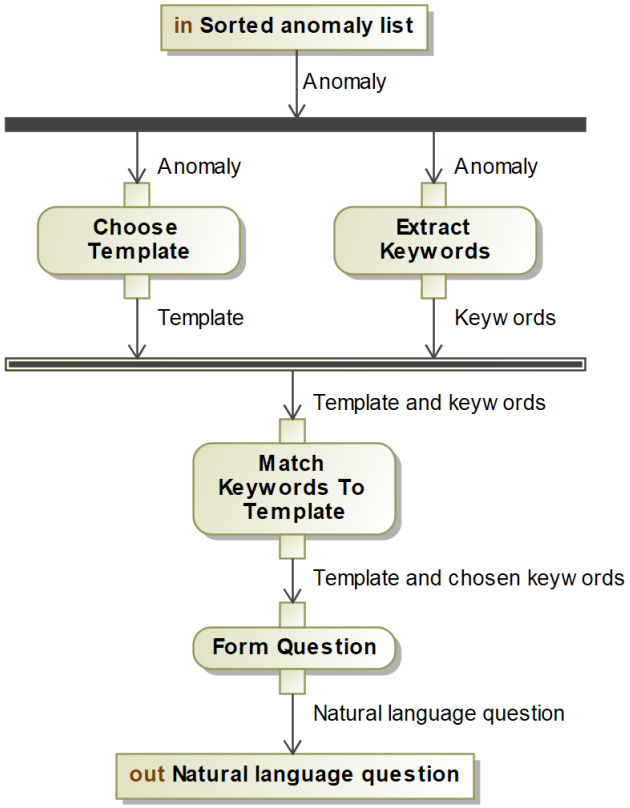

Figure 8. Forming a natural language question.

4.4. Analysing the User's Response

Once the user has responded to the query, we can begin analysis of the user's response, to determine if the user is authentic or fraudulent. Analysis will begin by reformatting the natural language response into a format more suitable to computer recognition, and then extract the most prominent keywords. The following component of AIAC's operation does not use neural networks, instead using more conventional methods to extract useful information from the user response.

The first thing to check for is user comprehension. If the user's response indicates that they did not understand the question, with a response such as "I don't remember that" or "what?" then analysis of the question ends there. If the user's response does not indicate

incomprehension, then AIAC proceeds to comparing the user response to the expected answer. For questions where the expected answer is a numeric value, AIAC can attempt to parse the user response as a numeric value and compare it to the expected answer to produce a similarity score. If the expected answer is "$20.55", then a user response of "Twenty bucks" would rate very similar to the expected answer. For questions where the expected answer is linguistic, a more complicated technique will be used to determine the degree of similarity between the expected answer and the user response.

The first step in comparing linguistic similarity will be to expand the expected answer and user response into a set of words similar to the expected answer and a set of words similar to the user response. This can be performed by looking up the expected answer and user response respectively in a thesaurus, and forming a set of words from the output of the thesaurus. The exact number of words chosen for the expected answer and for the user response is something to be determined through testing.

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:

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

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.

This equation places more emphasis on the number of matches found than the exact index of each match, but the indices of the matches are still tangibly relevant to the output score. When there are no matches found at all, the similarity score is determined to be 0.

The process of determining a similarity score for a linguistic response is illustrated in Figure 9, and the specific algorithm used can be described in Algorithm 1:

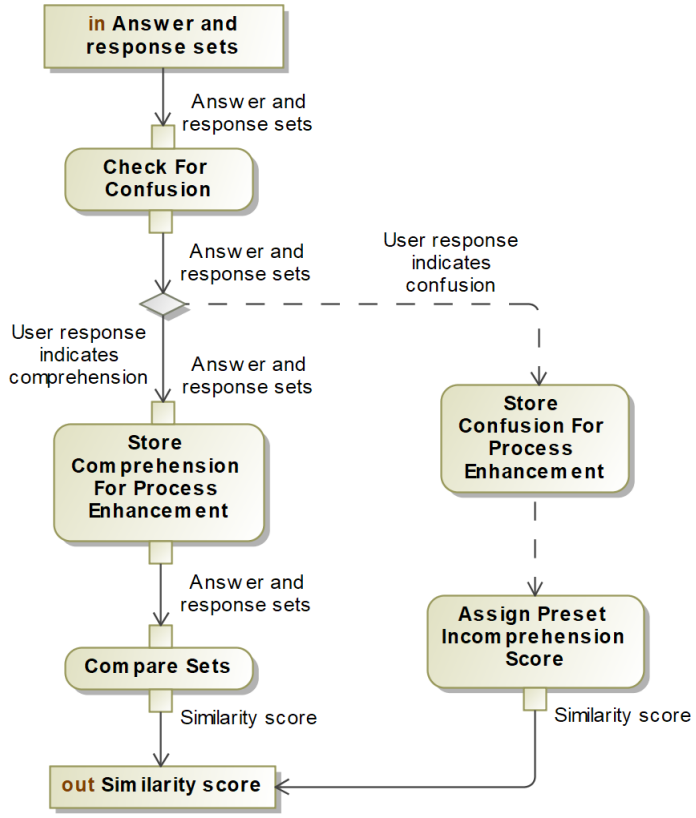

Figure 9. Analyzing the user's response.

Algorithm 1 Word-Matching Similarity Score Algorithm.

Input: The expected answer and the user response
Output: A number denoting the similarity between the two inputs

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.
for all inputs **do**
    Combine input with synonyms of input into a set of words.
end for
for all synonyms of expected answer **do**
    Find n synonyms of the synonym, adding them to the expected answer's list of words.
end for
for all synonyms of user response **do**
    Find m synonyms of the synonym, adding them to the user response's list of words.
end for
for all words in the expected answer's set of words **do**
    Compare word with each word in the user response's set of words.
    for all matches **do** Take the index of the matched word from both sets of words and pair them together.
    end for
end for
for all Index pairs gathered **do**
    Sum the two numbers together, then take the reciprocal of the sum.
end for
Sum up every reciprocal for output number.

The algorithm's first goal is to take each word, the user's response and the expected answer, and generate an expansive list of synonyms for each, ideally arranged in order of how similar they are to the original word. Once that has been done, the two sets of words are compared against each other, searching for cases where the same word has shown up in both lists. Since the synonyms are listed in order of how similar they are to the original word, the index of a word in its list is meaningful information to keep track of. Therefore, when we find a match, we record the indexes that the words were found at. Once all the matches have been found, we must find some way to translate them into a single numerical value that we can use. Because the synonyms are ranked in order of decreasing similarity, a lower index value suggests a closer relationship with the base words we are comparing. To capture this relation, we sum the two index values of each match together, creating a single numerical value for each match in which the lower the number the more significant the connection is. Since we want our output value to increase the more similar the words are, we take the reciprocal of each match's numerical value, creating numbers which are higher the stronger a connection they have to the base words. Then, we simply sum the numbers from all the matches together to get the aggregate similarity score between the two words.

However the analysis concludes, AIAC then moves on to attempting to determine if the user is authentic or fraudulent. AIAC will have a threshold for acceptance and a threshold for rejection, and if the user's cumulative similarity score crosses either threshold then AIAC will take the respective action. In cases where the user indicates incomprehension, a preset small negative similarity score will be used. If neither threshold is crossed, AIAC will choose the next-highest-rated anomaly on the sorted anomaly list and begin creating the second authentication challenge to present to the user. The similarity score from subsequent questions will be added to the cumulative similarity score until either a threshold is crossed or a certain number of questions have been asked, at which point AIAC will default to considering the user fraudulent.

The operators of AIAC will be able to adjust the thresholds AIAC uses to accept and reject people in order to tailor AIAC to their specific needs. A more restrictive threshold for rejection would more quickly eliminate fraudulent users, but also carries the risk of eliminating authentic users accidentally. A more restrictive threshold for acceptance would make it harder for a fraudulent user to guess their way to acceptance, but also increases the amount of questions an authentic user would have to ask in order to be accepted.

The ideal thresholds can be informed by activity logs created from user activity. By examining user authentication logs on a question-by-question basis, it would be possible to estimate the effects a given threshold would have on the users, such as how many questions an authentic user would expect to have to answer or how many authentic users might be accidentally considered fraudulent.

5. Experiments

In order to demonstrate the foundational principles of AIAC, sample data have been acquired from anomaly profiles generated by a program which detects anomalous activities from recorded activities and compiles them into anomaly profiles distinct for each person. The dataset consists of 619 anomalies spread across 20 anomaly profiles. The data samples allow us to, in absence of a functioning AIAC system, construct scenarios that replicate the desired features in order to prove the viability of the technologies.

The first experiment emulates the portion of AIAC that tracks whether the chosen anomaly was useful for distinguishing between authentic and fraudulent users and improves itself so as to apply that knowledge to pick the most useful anomalies in the future.

A series of sample questions were created, tailored to specific anomaly types and requesting a specific type of information for its answer. These questions are then manually assigned to each anomaly in the dataset, emulating the question-formation portion of AIAC.

Next, each anomaly is assigned a random label, designating whether it is to be answered authentically or fraudulently. This is because, in absence of authentic and fraudulent users to observe the behaviour of, the answers must be created manually. To simulate au-

thentic answers, the 'user response' feature for a given anomaly is answered with reference to the details of the anomaly, with small errors such as rounding numeric values or the small chance of giving an incorrect textual answer. To simulate fraudulent answers, the 'user response' feature for a given anomaly is answered without reference to the details of the anomaly, using only general qualities of the anomaly profile such as the common ranges of its values to determine a rough range of values to guess within.

Once the simulated user response is generated, it is compared against the expected answer for the anomaly to create a new feature that marks the divergence between the expected answer and simulated response. For numeric values, such as quantities of money or dates, the divergence is the absolute value of the difference between the two values. For questions answered with words, an exact match was designated a divergence of 0, and 1 otherwise. For each category of answer type, divergence values are normalized based on the divergence values found within that category.

Once the divergence is calculated, it is compared to the authenticity label to determine how successful the anomaly was. In accordance with the principle that an authentic user should be able to always correctly answer their questions and a fraudulent user should never be able to correctly answer their questions, an anomaly labelled authentic calculates its success feature as 1—divergence, whereas anomalies labelled fraudulent simply uses the divergence value.

With the data properly labelled, we then create a neural network to learn from the data, using the Python API Keras. The neural network is feedforward with two deep layers, each containing twice as many nodes as the input layer so as to allow a reasonable degree of higher-order learning to take place. The output layer is a single value, predicting the 'success' feature of the anomalies, which can be used to rate newly gathered anomalies on how likely they are to be successful or not.

A total of 80% of the anomalies in the dataset were used to train the neural network, and the remaining 20% of the anomalies were used to test it. After training, the neural network could predict the success of the anomalies with an accuracy of 71.4%, which demonstrates that it will be possible for AIAC to use the results of its prior operation to learn how to choose useful anomalies from an anomaly portfolio to present to the user.

ROC curves [19] are another valuable way of visualizing the system's capability to distinguish authentic from fraudulent material. Figure 10 shows an ROC curve that compares the neural network's predictions against the labels it was predicting, and the result has a smooth curve and an area-under-curve of 0.96, indicating solid performance of AIAC.

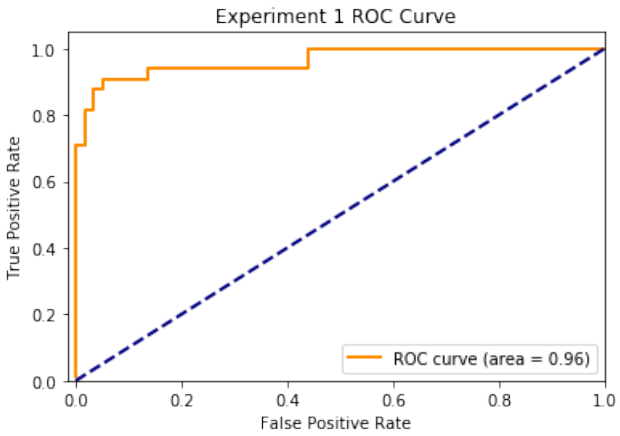

Figure 10. Experiment 1 ROC curve.

The second experiment emulates the portion of AIAC that gathers knowledge on which anomalies are comprehensible with which templates and improves itself so as to

apply that knowledge to make better template selections. To achieve this, we simulated the process of determining whether a posed question would be comprehensible or not.

Six templates were made, such that for every anomaly there would be at least one fully comprehensible question, and each anomaly was manually given a comprehensibility rating for each possible question.

The sample authentication challenge templates were as follows: "You spent an unusually large amount of money on a specific service recently. What was it?", <action type>, "You spent an unusually small amount of money on" <action type> "recently. When?", <time>, "What activity did you do recently instead of", <expected activity>, "?", <unexpected activity>, "You went to a different store than usual recently. When did you do this?", <Time comparison>, "You went to", <activity type>, "earlier than usual. When would have been normal?", <time>, and "You went to", <activity type>, "later than usual. When would have been normal?", <time>.

The neural network constructed to learn from the input data is of a similar structure to the neural network used for the first experiment, also constructed using the Keras API, except for the fact that the output layer has multiple nodes, one for each question. The neural network is tasked to predict all of the comprehensibility labels for each anomaly.

A total of 80% of the anomalies in the dataset were used to train the neural network, and the remaining 20% of the anomalies were used to test it. After training, the neural network could predict the comprehensibility of a given anomaly for each of the possible question formats with an accuracy of 84.0%. This shows that AIAC will be able to learn which sentence templates are most suitable to the chosen anomaly for the purpose of constructing a comprehensible question.

The final experiment emulates the word-matching algorithm used to compare user response to expected answer. To model this situation, as we are lacking authentic user responses to utilize for comparison and the anomaly profiles will not provide us with what we need, we instead manually created a list of 80 sample response/answer pairs, split evenly into the following categories: (i) exact matches, (ii) mostly correct responses, (iii) mostly incorrect responses, (iv) incorrect responses.

We expand each word into a list of words by finding each word's synonyms using PyDictionary, a python module which connects to synonym.com to find a given word's synonyms. For any one synonym lookup, no more than the 10 top synonyms are recorded, if more exist, so that words with many synonyms are not unduly privileged. One degree of separation is not enough, however, and so we also gather the synonym of synonyms by applying PyDictionary to each synonym in the synonym list. No word is added to the list of words more than once, to ensure that the matches found are unique. This process can be extended to as many degrees of separation as is needed, and four degrees of separation to show a relatively wide search.

With the full word lists assembled for the two words to be compared, every word in the first list is compared against every word in the second list, recording all matches found as pair-values encoding the index of the matched words in their respective lists. The collected pair-values are then processed into a single output similarity value with a sum of reciprocal values equation, to determine how similar the words are.

The results of this experiment from the 80 sample answer/response pairs are graphed on Figure 11, and the average values across type of response is shown in Figure 12:

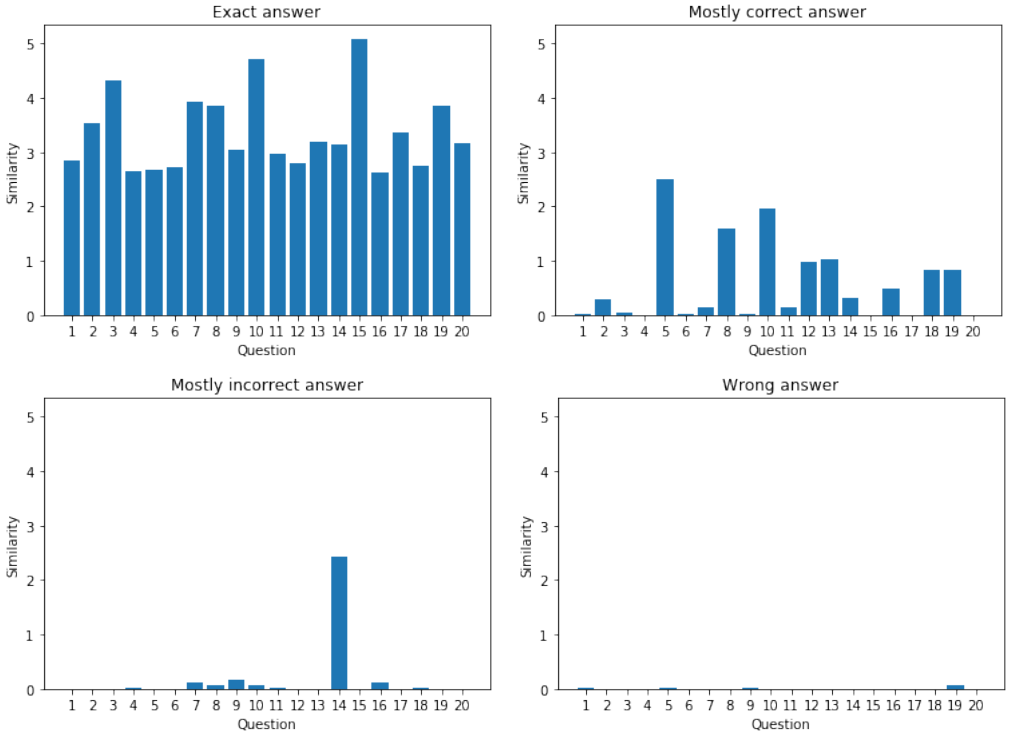

Figure 11. Results of word matching implementation.

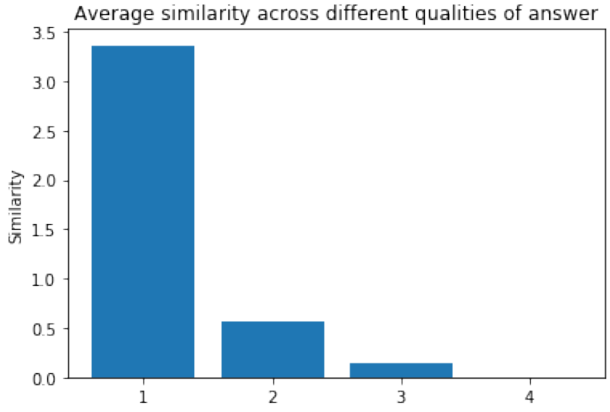

Figure 12. Average word matching similarity.

What we can see here is a noticeable gradient in average similarity between the different classes of actions, with the better matches having a higher expected similarity than the worse matches. We can also see that there is a problem of even mostly correct responses not finding any match at all, but the odds of this happening decrease the greater the degree of separation used.

In addition, we prepared three more ROC curves, shown in Figure 13, dividing our dataset into different binary groups for the sake of analysis.

The first ROC curve compares 'Exact Answer' against the other three classes. Since all exact matches inherently achieve higher scores than other classes, the ROC curve is flat line with an area-under-curve of 1.00. The second ROC curve compares 'Exact Answer' and 'Mostly Correct Answer' against the other two classes, dividing the dataset in half. Its area-under-curve is 0.95 with a smooth curve, indicating only a small amount of overlap between the upper and lower halves of the dataset. The third ROC curve compares 'Incorrect Answer' against the other three classes. Its area-under-curve is 0.85, indicating significantly more overlap than the earlier curves but still remaining clearly distinguishable.

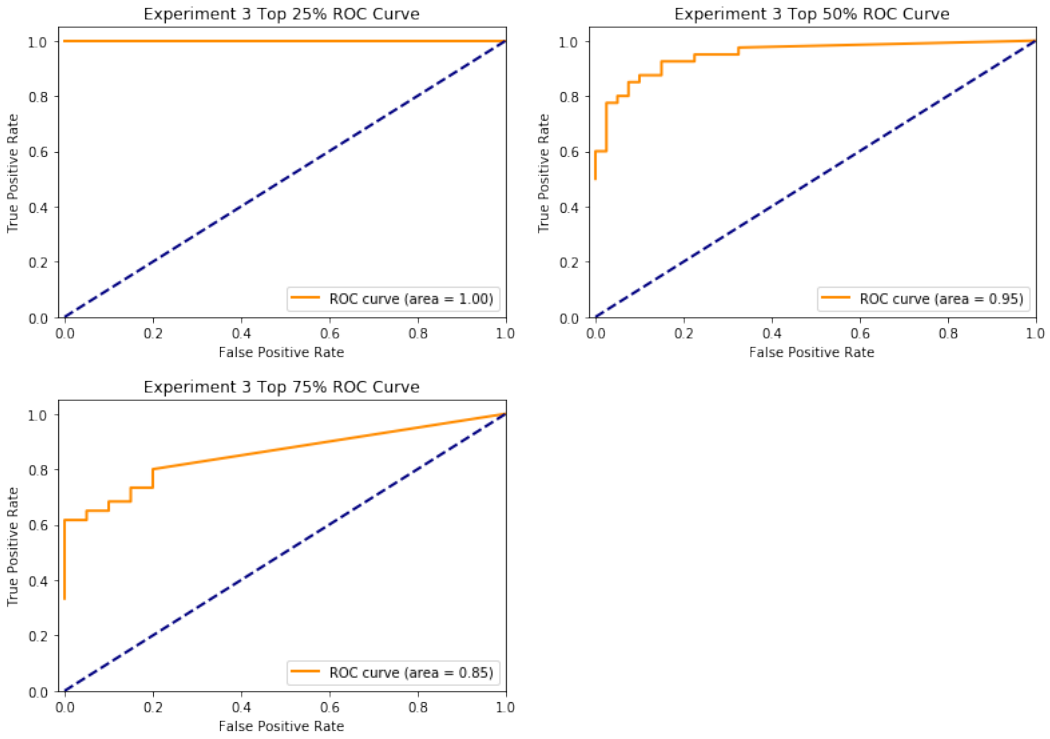

Figure 13. Experiment 3 ROC curves.

Overall, these results show that this structure of word matching holds strong potential for assessing how close a user's response is to the expected answer.

6. Conclusions

There exists a tradeoff between security and accessibility in common authentication methods, with methods such as passwords being highly accessible but performing poorly on security, and with methods such as biometric readings being highly secure but comparatively inaccessible.

A new authentication framework has been posed, which uses human behaviour in a JitHDA system to dynamically construct questions that only the authentic user would be capable of answering but do not hold the weaknesses of static passwords, so that when used in conjunction with passwords the overall security of the system will be greater without sacrificing accessibility.

AIAC is a proposed implementation of JitHDA that can achieve its desired goals by using recent anomalous events gathered from recorded user activity to create authentication questions that only the authentic user should be able to answer, with no need to remember anything other than recent memorable events in their life.

AIAC uses a neural network to choose anomalies that authentic users are likely to get right and fraudulent users are likely to get wrong, and uses a second neural network to choose a template to use to pose the anomaly as a question to the user. Once the user responds, their response is used to help determine if they are an authentic user or not, and the data gathered is then used to help AIAC improve itself.

We assembled three test experiments that emulate critical portions of AIAC's functionality: the neural network that choose anomalies, the neural network that chooses question structure, and the algorithm that compares word similarity. In each experiment, the results proved favourable to the idea of an expanded implementation, as might be seen in a fully realized AIAC, being able to properly fulfil the desired functionality. As a result, we can confidently say that the goal of proposing a system capable of fulfilling JitHDA has been satisfactorily accomplished in this paper.

The next step for fulfilling JitHDA would be to create a full holistic implementation of AIAC within a real industry setting, to evaluate how accurately system performs in practice. This will also allow for the fine-tuning of various elements of AIAC's design so as to maximize its ability to discern between authentic and fraudulent users. It is also possible to consider alternate implementations of some of AIAC's features, such as the word-matching mechanism used to create similarity scores or how sentence templates are structured and constructed.

Author Contributions: Conceptualization, A.O.; methodology, P.V.; software, P.V.; validation, P.V.; formal analysis, P.V.; investigation, P.V.; resources, I.I.M.A.S.; data curation, I.I.M.A.S.; writing—original draft preparation, P.V.; writing—review and editing, P.V.; visualization, P.V.; supervision, A.O.; project administration, A.O.; funding acquisition, A.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) [Grant no. RGPIN-2018-06250], and Taif University Researchers Supporting Project [Number TURSP-202é145], Taif University, Taif, Saudi Arabia. These supports are greatly appreciated.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in the creation of this paper can be found at the following sources: https://www.kaggle.com/datasets/ealaxi/banksim1, https://github.com/atavci/fraud-detection-on-banksim-data, https://www.researchgate.net/publication/265736405_BankSim_A_Bank_Payment_Simulation_for_Fraud_Detection_Research.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Ashibani, Y.; Mahmoud, Q.H. A multi-feature user authentication model based on mobile app interactions. *IEEE Access* **2020**, *8*, 96322–96339. [CrossRef]
2. Wong, A.B. Authentication through Sensing of Tongue and Lip Motion via Smartphone. In Proceedings of the 2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Rome, Italy, 6–9 July 2021. [CrossRef]
3. Mohamed, N.; Al-Jaroodi, J.; Jawhar, I.; Kesserwan, N. Data-driven security for smart city systems: Carving a trail. *IEEE Access* **2020**, *8*, 147211–147230. [CrossRef]
4. Raponi, S.; Pietro, R.D. A Longitudinal Study on Web-Sites Password Management (in)Security: Evidence and Remedies. *IEEE Access* **2020**, *8*, 52075–52090. [CrossRef]
5. Ouda, A. A framework for next generation user authentication. In Proceedings of the 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, Oman, 15–16 March 2016; pp. 1–4.
6. Abu Sulayman, I.I.M.; Ouda, A. Designing Security User Profiles via Anomaly Detection for User Authentication. In Proceedings of the International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 20–22 October 2020.
7. Liu, B.; Xu, Z.; Sun, C.; Wang, B.; Wang, X.; Wong, D.F.; Zhang, M. Content-oriented user modeling for personalized response ranking in chatbots. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 122–133. [CrossRef]
8. Kao, C.; Chen, C.; Tsai, Y. Model of multi-turn dialogue in emotional chatbot. In Proceedings of the 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Kaohsiung, Taiwan, 21–23 November 2019; pp. 1–5.
9. Patel, F.; Thakore, R.; Nandwani, I.; Bharti, S.K. Combating depression in students using an intelligent chatbot: A cognitive behavioral therapy. In Proceedings of the 2019 IEEE 16th India Council International Conference (INDICON), Rajkot, India, 13–15 December 2019; pp. 1–4.
10. Srivastava, P.; Singh, N. Automatized medical chatbot (medibot). In Proceedings of the 2020 International Conference on Power Electronics IoT Applications in Renewable Energy and Its Control (PARC), Mathura, India, 28–29 February 2020; pp. 351–354.
11. Zhu, T.; Qu, Z.; Xu, H.; Zhang, J.; Shao, Z.; Chen, Y.; Prabhakar, S.; Yang, J. Riskcog: Unobtrusive real-time user authentication on mobile devices in the wild. *IEEE Trans. Mob. Comput.* **2020**, *19*, 466–483. [CrossRef]
12. Kim, S.; Kim, S. General authentication scheme in user-centric idm. In Proceedings of the 2016 18th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Republic of Korea, 31 January 2016–3 February 2016; pp. 737–740.
13. Dostálek, L. Multi-factor authentication modeling. In Proceedings of the 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 5–7 June 2019; pp. 443–446.

14. Tuna, T.; Akbas, E.; Aksoy, A.; Canbaz, M.A.; Karabiyik, U.; Gonen, B.; Aygun, R. User characterization for online social networks. *Soc. Netw. Anal. Min.* **2016**, *6*, 104. [CrossRef]
15. Gahi, Y.; Lamrani, M.; Zoglat, A.; Guennoun, M.; Kapralos, B.; El-Khatib, K. Biometric identification system based on electrocardiogram data. In Proceedings of the 2008 New Technologies, Mobility and Security, Tangier, Morocco, 5–7 November 2008; pp. 1–5. [CrossRef]
16. Sulayman, I.I.M.A.; Ouda, A. User modeling via anomaly detection techniques for user authentication. In Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 17–19 October 2019; pp. 0169–0176.
17. Sulayman, I.I.M.A.; Ouda, A. Human Trait Analysis via Machine Learning Techniques for User Authentication. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications(ISNCC), Montreal, QC, Canada, 20–22 October 2020.
18. Lopez-Rojas, E.A.; Axelsson, S. BankSim: A Bank Payment Simulation for Fraud Detection Research. In Proceedings of the 26th European Modeling and Simulation Symposium, Bordeaux, France, 22–24 September 2014.
19. Prati, R.C.; Batista, G.E.A.P.A.; Monard, M.C. Evaluating classifiers using roc curves. *IEEE Lat. Am. Trans.* **2008**, *6*, 215–222. [CrossRef]