2021

An Online neural network based password prediction, generation, and storage scheme.

Mbaka, Winnie Bahati
Faculty of Information Technology
Strathmore University

An Online Neural Network Based Password Prediction, Generation, and Storage Scheme

Mbaka, Winnie Bahati

Submitted in partial fulfilment of the requirements for the Degree of Masters of Science in Information Systems Security at Strathmore University.

Strathmore University

Faculty of Information Technology

Nairobi, Kenya

September, 2021

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Student Mbaka Winnie Bahati

Signature

Date 9th September, 2021

APPROVAL

This dissertation of Mbaka Winnie Bahati, was reviewed and approved by:

Dr. Humphrey Njogu,
Lecturer, Faculty of Information Technology,
Strathmore University

Dr. Julius Butime,
Dean, School of Computing and Engineering Sciences,
Strathmore University

Dr. Bernard Shibwabo,
Director of Graduate Studies,
Strathmore University

Abstract

The gradual change from traditional workplaces to online platforms has been attributed to shifting user requirements, economic factors, and lifestyle differences. Perhaps the most significant factor attributed to this change may be the advent of the 2019 outbreak of the Coronavirus pandemic making the topic of physical interaction among some of the severely affected aspects of life. To remedy this situation, all knowledge and employment institutions adopted various online platforms as a means of maintaining a continued learning and working processes.

However, these technical advances presented the issue of upholding information integrity of individuals accessing materials over the Internet as they were required to authenticate themselves prior to gaining access to secured resources. However, authentication processes such as the use of passwords are prone to guessing attacks, one of the biggest challenges in modern computing. Such attacks occur because of the vulnerabilities of human-chosen passwords. Research indicated that despite innovation on other safer authentication mechanisms, passwords continue to dominate the authentication space because they are memorable, free and user-generated.

In view of the above shortcomings, this study sought to develop an online scheme that is geared towards helping Internet users, generate stronger passphrases based on how predictable their preferred passwords are. To understand the underlying technologies in the creation of stronger passwords, the study analysed existing literature on the character composition of human-created passwords and available tools that can be used to perform predictive analysis and generation of complex secret words. Additionally, password managers were studied to realise their functionality in securely storing complex passphrases.

Analysis of the findings of the research asserted the need to incorporate neural networks, integrated data-driven insights, and derived concepts from the Markov chain model in the development of an online password predictive and generative scheme with an embedded password manager that allowed users to store the complex secret words. The resulting accuracy score after the scheme was trained using 50 epochs stood at 0.90332413, equivalent to 90.3%.

KEYWORDS

Passwords, Neural Networks, Markov Chain Model, Predictive Analysis

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2FA: Two Factor Authentication

AES: Advanced Encryption Standard

AI: Artificial Intelligence

ANN: Artificial Neural Networks

ASCII: American Standard Code for Information Interchange

CPU: Central Processing Unit

CSV: Comma-separated values

HTML: Hypertext Mark-up Language

JtR: John the Ripper

LSTM: Long Short-Term Memory

ML: Machine Learning

MFA: Multi-Factor Authentication

PC: Personal Computer

PE: Processing Elements

PII: Personal Identifiable Information

PIN: Personal Identification Number

RAD: Rapid Application Development

RNN: Recurrent Neural Networks

SGD: Stochastic Gradient Descent

SSL: Secure Socket Layer

SFA: Single Factor Authentication

URL: Uniform Resource Locator

Chapter 1: Introduction

1.1 Background of the Study

Despite the introduction of more secure biometric devices, passwords have remained the most common authentication method. Unfortunately, research indicated that passwords are prone to various forms of attacks that compromise people's security and privacy, mainly because passwords are heavily dependent on a human's ability to remember (Farash, 2015). Some of the widely researched security threats against online platforms include denial of service, session hijacking, brute-force attacks, information exposure, and spoofing identity. According to Chopra (2016) the impact of security breaches on online platforms has resulted in significant damages on the various stakeholders that interact with such applications. More importantly, the effectiveness and trust levels of continued use of such platforms by the general public becomes questionable.

Primarily, studies have indicated that there is a huge discrepancy between required secured entropy and the one that human brain can remember. For instance, a 16-character password with numbers, punctuation, lowercase and uppercase letters would take more than 1 trillion years to crack (Mazurek, 2013). However, human beings can only remember less than seven pieces of information of that password at once. Furthermore, when users are forced to use the 16-character password, they will note it down and store them in vulnerable places that can be accessed by potential attackers (Mazurek, 2013). Lastly, users are likely to memorize one secure password and use it all their accounts. Using one password for many accounts makes it more valuable to the attacker. Based on these findings, there was need to evaluate the efficacy of machine learning techniques in developing a tool that can perform predictive analysis on human created passwords while providing a means by which users can generate and subsequently store complex secret words.

In a 2013 research titled Rethinking Passwords by William Chestwick, an observation was made with regards to machine generated passwords. The author (2013) noted that such passwords presented a much stronger security against hackers compared to human created secret words. Chestwick's findings were supported by those of Komanduri et al. whereby heuristic password guessing algorithms were investigated alongside the relationship between password entropy estimates and password-guessing algorithms (Komanduri, 2012).

1.2 Statement of the Problem

Internet users tend to create passwords that exhibit some patterns and set of characteristics, such as including personal information. It is a common observation that individuals prefer to keep their passwords simple for ease of remembrance. This is because, most online interactions require a form of user authentication. Keeping up with multiple passwords can be difficult to remember, resulting in the creation of simple passwords that have a predictable pattern. Such passwords are usually very vulnerable to guessing attacks. In a survey conducted by Kaspersky, it was discovered that there has been a surge in the number of distributed denial of attacks that occurred in the first quarter of 2020 (Kaspersky, 2020). As the popularity of online platforms continues to grow, the threats against such applications are expected to increase.

1.3 Research Aims and Objectives

The main objective of this study was to develop an online scheme that sought to predict human-created passwords letter-by-letter to completion, in real time, then suggest a stronger computer-generated passphrase within set parameters, such as preferred length, symbols, numbers and lower or uppercase letters. Due to the complexity of the generated password, a password manager will be embedded into the scheme for the purpose of securely storing the new secret word.

1.3.1 Specific Objectives

The specific objectives are:

I. To understand the character composition of passwords, factors that predisposes passwords to guessing attacks, and the use of password managers in storing passwords.

II. To understand previous research in password predictive analysis and how machine learning techniques can be used to enhance password creation.

III. To design, develop and test a neural network model that is able to perform predictive analysis on user created passwords, generate complex keywords and store them within a password manager.

IV. To validate that predictive analysis on password can be used to improve character composition of human-created passwords.

1.3.2 Research Questions

The following questions helped to achieve the objectives of this study:

I. What are the common character composition of passwords, factors that predispose passwords to attack, and how can password managers be used to store passwords?

II. What research has been previously done on password prediction and how can machine learning techniques be leveraged to improve password creation?

III. How can a neural network scheme be developed and trained to predict human created passwords, generate complex secret words and securely store them?

IV. Can predictive analysis be used to improve the character composition of human-created passwords?

1.4 Justification of the Research

Given the current increase in the usage of online platforms globally, hackers have targeted multiple online portals in an attempt to steal users' personal information. Various security researchers have discovered numerous unprotected databases containing personally identifiable information (PII) of millions of users including, names, passwords, emails, and financial information (Williams, 2021). The compromised data can be used by malicious persons to commit several cybercrimes, such as identity theft, blackmailing, bullying, stalking, and phishing scams.

1.5 Scope and Limitations

This study focused on how Internet users construct their passwords and their composition policies. The analysis was done on commonly preferred passwords to create a scheme that can predict human chosen passwords. Based on the ability of the scheme to perform predictive analysis, a stronger, more complex password was generated which maintained user specified parameters, such as length. The complex password was stored within an embedded password manager. The aspects that were examined in this study include, character combinations and patterns of human-created passwords, predictive analysis, generation techniques, and password storage systems. This study made the following assumptions:

I. Passwords are not 100% secure

II. Human created passwords are prone to guessing attacks such as brute force dictionary attacks

III. Machine learning techniques can be used to strengthen password creation

With the existence of multiple password-predicting models, this research was limited to the use of artificial neural networks. The model was developed using Python programming language while the password manager relied on Vuejs for the front-end and Laravel framework for the back-end and limited to the Windows environment although testing and validation was also performed on

other operating systems browsers. The scheme enforced heuristics such as character sets and variables that are resistant to brute-force and dictionary attacks.

1.6 Summary

Password based authentication methods continue to dominate both online and offline applications. In this case, Internet users are required to create a password for most of all their online interactions. This phenomenon has resulted into an instance whereby users are more inclined to using a simpler password that contain certain linguistic characteristics which make their passphrases easier to remember. The preference of simplicity as opposed to complex and more secure passwords have increased the number of malicious attacks that are aimed at gaining information that can either be sold or used to blackmail the owners.

Chapter 2: Literature Review

2.1 Introduction

Computing and Internetworking are branches of technology that are growing at a profound rate. The growth created the need for the implementation of authentication protocols prior to access computing and Internetworking systems. People use different authentication protocols such as passwords, PINs and biometrics to secure their information in these systems. Research indicated that passwords are the most dominant form of authentication because they are memorable, free and user-generated. However, despite the advantages, passwords remain the most venerable form of user verification (Fang, 2019). As a result, passwords have increased the probability of privacy and security of millions of users being compromised. Therefore, this chapter evaluated different scholarly works published on password vulnerability, and how machine learning techniques can be used to address the problem.

2.2 Character Composition of Passwords

Jakobsson and Dhirman's (2013) study was an evaluation of the character composition of passwords. The objective of the study was to create an outline of what constitutes a strong password. The authors indicated that data on properties of passwords could either be obtained from leaked datasets or surveys. Furthermore, the authors noted that the properties of passwords could be based on the policies of the password, according to their research, password policies can be classified into two: simple or complex.

The simple policy represents a scenario whereby the length of a password requires eight characters or less and at most one unique character. Complex policy on the hand is when the requirements of a password are strict. The findings of the authors (2013) study revealed that passwords of systems that use simple policies are the most vulnerable to attacks. For instance, the authors studied that commercial websites are prone to password guessing attacks because they follow simple policies. Some of the most commonly used passwords on commercial websites include Iloveyou, angels, princess, and password1.

On the other hand, complex policies are used to generate passwords that consume a lot of time and resources in the event of an attack. Primarily, complex policies encourage the creation of passwords that exceed eight characters consisting of symbols, numbers, punctuation, lowercase and uppercase letters (Jakobson, 2013). However, the authors highlighted that under complex policy there are individuals who create new passwords by modifying their previous

passwords. Most transformations include changing one symbol and or moving digits. Modified passwords can be cracked in less than five guesses if the original one is known.

Additionally, the authors revealed that there are linguistic elements that determine the strength of passwords. Their analysis indicated that passwords contain linguistic elements such as keyboard patterns, dictionary words of names of people and places. Such linguistic elements are intended to make passwords more memorable. However, the extensive use of names of places create effective guessing dictionaries for potential attackers. Therefore, the authors (2013) concluded their study by stating that strong passwords are the ones that adhere to complex policies. In this case, a strong password should be a mixture of symbols, numbers, punctuation, and lowercase and uppercase letters. Furthermore, individuals should avoid linguistic elements such as names of places while creating passwords. These elements create guessing dictionaries for potential attackers. One of the major disadvantages of using various complex passwords is that users are likely not to remember. Hence, researchers, administrators, and other password security stakeholders recommend the use of password managers.

The results of the authors are consistent to those of Mazurek et al. (2013) who found that the demand to remember passwords encourages linguistic elements such as names of people or places. In their analysis, Mazurek (2013) found that most of the passwords that users in their study selected were bigrams or linguistic elements that could be found on Google Web and British National Corpus. In most cases, the passwords were movie or book titles. The linguistic elements according to Mazurek (2013) offered effective guessing dictionaries for potential attackers. Mazurek (2013) concluded that strong passwords should be created by avoiding linguistic elements such as movie and book titles or names that provide effective guessing dictionaries.

Ciampa (2013) revealed that password entropy is one of the most effective techniques used to measure password strength. By definition, password entropy is the measurement of how unpredictable passwords are based on their character set or combination and length. Ciampa (2013) further indicated that the entropy of a password is usually conveyed in terms of bits. For instance, a password that has zero bits of entropy is one that is already known. Ciampa (2013) stated that the strength of passwords could be calculated by identifying the entropy of each character. The entropy is calculated by multiplying the number of characters in a password with

the entropy of each character which is a log base 2 of the number of characters. Ciampa (2013) concluded by reiterating the high entropy passwords are hard to crack.

2.3 Factors that predispose Passwords to Attacks

Several studies have examined various factors that make it easier for hackers to gain access to secured platforms. Although majority of Internet users have some rudimentary knowledge on what constitutes a strong password, the extent of passphrase reuse is an issue that needs to be addressed. A recent survey conducted in the United States indicated that password reuse is a common practice, especially among people aged between 18 and 24 (Lord, 2020). Although the reusability aspect was present in all ages, the prevalence reduced significantly in older people. With regards to reusability, Mazurek (2013) identified that using one password for multiple accounts created a highly-valuable target that can be exploited easily by attackers.

Mazurek (2013) study revealed that password guessing attacks are based on the limitations of human cognition making it the primary reason why users make weak passwords. Ideally, users are prone to make passwords that are easy to recall. As a result, users follow common patterns and combinations that can be predicted by attackers. Results from the survey conducted by Lord (2020), indicated 18 percent of the participants preferred simple passwords that are easy to remember.

Consequently, Mazurek (2013) revealed that when users are asked to memorise strong passwords they tend to note or store them in a place that can be accessed by a potential attacker. These findings were supported by Lord's survey whereby the majority of participants, 39%, indicated that they write their passwords on pieces of paper. 10 percent store them in a file on their computers, 7% uses drobox services, while the remaining 28% use secure password managers.

Extensive research indicates the benefits of changing passwords regularly, making it one of the requirements when implementing complex password policies in organisations. The findings of Lord (2020) indicated that this is usually not the case. 11% of the participants mentioned that they have never changed their passwords while 18.5 % are inclined to modify their passphrase only when notified about a security issue.

Although studies from both authors, Mazurek (2013) and Lord (2020) explain the concept of password guessing and why human-created passwords are prone to guessing attacks. Their research was however, limited to online password creation policies while there is an increasing prevalence in offline attacks.

2.4 Password Managers

Password managers are application software used to securely encrypt and store user passwords. According to Jancis (2021), a large number of cybersecurity experts suggest that password managers are currently the most secure way to protect one's passphrases. However, extensive use of these applications is not a guarantee of a hacker-free online environment. Password managers use several ways to ensure the security of its content. First, such applications require a master password that secures the users account, otherwise known as a vault (Jancis, 2021). The information in the vault is encrypted by implementing a zero-knowledge architecture. The encryption is based on the advanced encryption standard (AES) 256-bit technology which is also implemented by the military because of its difficulty in cracking the resulting cipher (Jancis, 2021). There are different types of password managers based on their technology, including cloud-based, browser-based, and desktop applications. Google password manager is among the most widely used password managers.

2.4.1 Google Password Manager

Google password manager is a web based application that is embedded into the Chrome browser. This application is particularly beneficial to online users who do not require extensive functionality (Password manager, 2021). However, non-android users who may require the use of password managers, there are other services, such as Lastpass or 1password that provide additional services at a fee. Table 2.1 highlights the various advantages and disadvantages of using Google password manager.

Pros	Cons
Accessible on any device that support chrome browser	Does not have an upgrade option for additional services
Free resources, does not require a subscription	Not compatible with other browsers
Auto-fills and saves passwords	Its functionalities are browser specific

Table 2.1: Pros and Cons of Using Google password manager

2.4.2 LastPass Password Manager

Passwords that are stored in LastPass are protected using the 256-bit advanced encryption system (AES) algorithm. To access stored passwords, users are expected to set-up a master passphrase

whereby LastPass has implemented a security challenge approach which reviews ones' passwords informing them whether or not their passphrases are strong or not (Martindale, 2021). LastPass offers a multi-factor authentication process similar to the ones implemented in Google password manager. Unfortunately, LastPass has been compromised in the past whereby a software bug allowed malicious attackers to access user passwords (Keane, 2017). However, the issue has since been fixed.

2.4.3 1Password Password Manager

Compared to LastPass which stores its encrypted passwords in a remote server, 1Password maintains its passphrases locally. However, both managers implement the 256-bit AES encryption algorithm. 1Password employs the use of a secret key, along with a master password during login sessions (Martindale, 2021). This password manager does not involve third party authentication systems, however, it incorporates biometric fingerprint scanning as an optional 2-factor authentication. One weakness observed while using 1Password is that it provides an emergency file that can be stored digitally or printed (Martindale, 2021). This file contains all the users' credentials in plain text, a phenomenon that shifts the burden of securely storing the file back to the user.

2.5 Machine Learning Based Techniques

Several different techniques have been developed over the years to protect users against identity and account theft. These techniques however, are not used widely mostly because users do not appreciate the added steps of the authentication process (Dickson, 2018). According to Dickson (2018) current authentication schemes have not been keen on maintaining the sensitivity and the value of users' passwords. This can be observed by the number of usernames and passwords that have been sold on the dark web. Studies (4iQ, 2017) uncovered a 41GB file containing 1.4 billion combinations of username and passwords from online platforms such as Netflix and LinkedIn that were being sold in the dark web.

To strengthen password creation schemes, Dickson (2018) proposed the use of Artificial Intelligence (AI) and Machine learning (ML) techniques. The growing availability of data and connectivity provides plenty of ways to manipulate and use ML technologies to realize their full potential. However, such technologies rely heavily on availability of large sets of data.

2.6 Artificial Neural Network

Artificial Neural Networks (ANN) according to Dell'Amico (2010) represents machine-learning techniques that are designed to model human neurons. Della'Amico (2010) stated that neural networks are brain-inspired systems that are designed to copy how human beings learn. Therefore, neural networks are effective tools for finding complex patterns. Since neural networks are brain-inspired, see figure 2.2, they can extract complex patterns that are numerous for a human programmer.

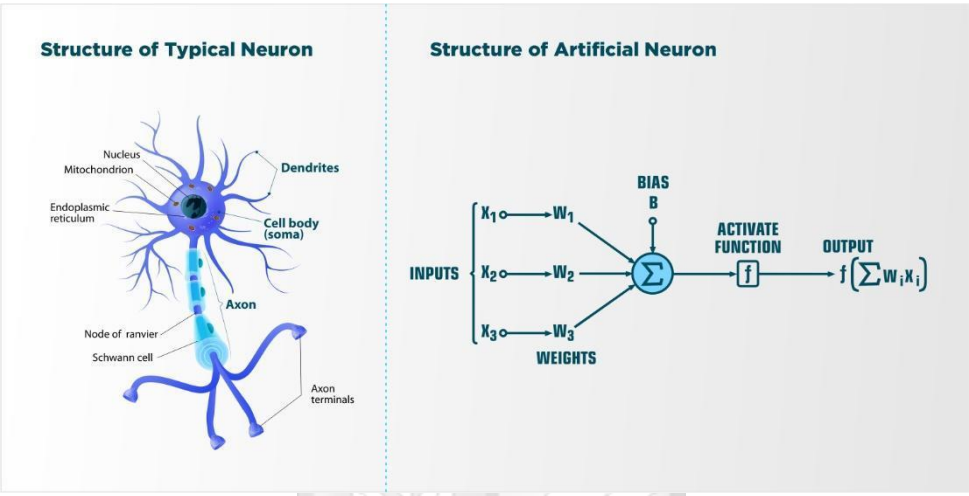


Figure 2.1: Brain Inspired Neural Networks

Source (Aggarwal, 2020)

Artificial neural networks have two main parameters that control the architecture of the entire network (Aggarwal, 2020);

- The number of layers
- The number of nodes in each hidden layer

A typical neural network has three distinct layers: an input, hidden, and an output layer. The input layer is used to feed the source data to the network. The input layer is followed by several hidden layers, depending on the implementation of the neural network. Hidden layers are designed to produce a specific output to an intended result. Figure 2.3 is a diagrammatic representation of layers in a typical neural network.

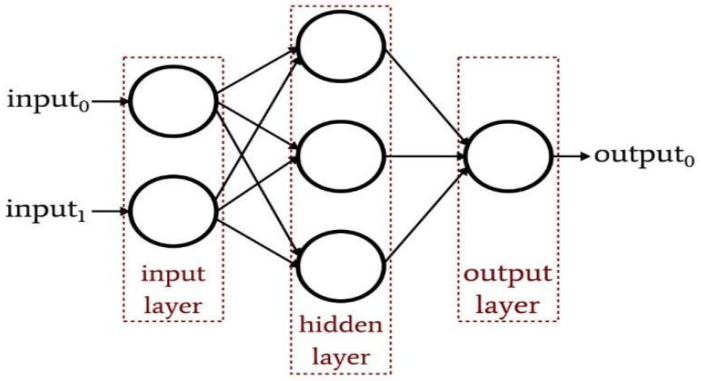


Figure 2.2: Artificial Neural Networks Layers

Source (Keim, 2020)

When determining the number of hidden layers that will go into the model, a few considerations should be taken into account. The considerations are summarized in the table 2.2.

No. of hidden layers	**Result**
0	Models with no hidden layers are only capable of representing linear separable functions
1	Models with one hidden layer are capable of approximating functions that contains a continuous mapping from one finite space to another
2	Models with two hidden layers are capable of representing arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy
>2	Models with more than two hidden layers can learn complex representations, sort of an automatic engineering feature

Table 2.2: Considerations for Determining Hidden Layers of the Model

2.6.1 Recurrent Neural Network

Recurrent Neural networks (RNN) are designed to work with sequence prediction problems. An RNN consists of internal loops which induce recursive dynamics in the networks that leads to delayed activation dependencies across the processing elements (PEs) in the network (Marhon, 2013). These kinds of neural networks process elements in sequences with the use of their internal memory to remember information about previous elements in the sequence, hence being a multi-layered neural network. Typically, the same function and parameters are used at every time step. The most successful type of RNN is the long short-term memory.

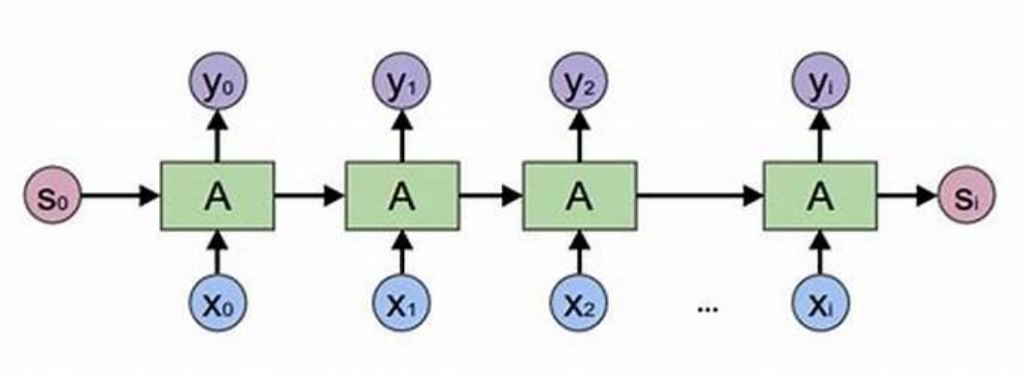


Figure 2.3: Recurrent Neural Network

Source (Olah, 2015)

2.6.2 Long-Short Term Memory

Long-short term memory (LSTM) is a recurrent neural network that has been used in natural language processing because it does not suffer from the vanishing gradient, an issue that arises from training machine learning algorithms through the gradient descent (DeepAI, 2019). All recurrent neural networks have the form of a chain of repeating modules of neural networks. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer (Olah, 2015).

Text generation is a language modelling that is at the core of natural language processing tasks such as speech to text, conversational systems and text summarization (Bansal, 2018). Long short-term memory architecture was developed to learn sequences of words from multiple inputs stories which constitutes of three major modules;

- Input processing- Takes a sequence of words as input
- Creating and training the neural network- computes the output using LSTM units, the units in an LSTM layer can be defined beforehand.
- Generating an output- Computes the probability of the best possible next word as the output.

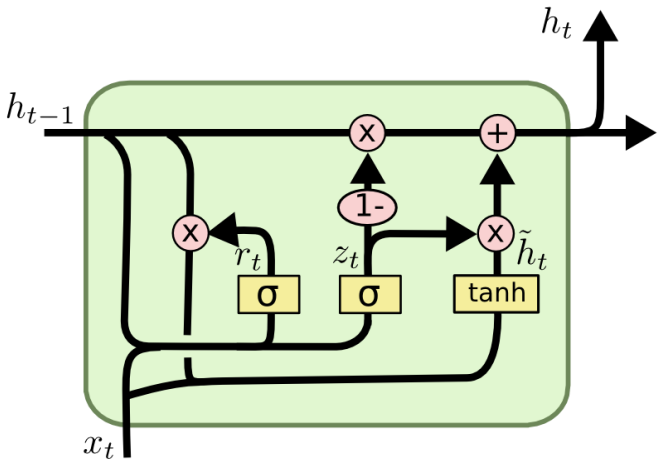


Figure 2.4: Long Short-Term Memory

Source (Olah, 2015)

2.7 Markov Model

The Markov model is considered a statistical model that is instrumental in performing predictive analytics especially those that rely on a probability theory (Plötz, 2011). Being an un-précised model, Markov is mostly used in systems that do not follow any fixed pattern, that is systems with randomly changing variables. The Markov model is based on the premise that future occurrences are completely dependent on current state. In the case of password prediction, the next character depends on the previously inputted text (Plötz, 2011). The proposed scheme will be based on the concepts of the Markov model with regards to performing its predictive analysis. These concepts can be divided into three distinct steps as seen in figure 2.6, including input, processes, and output.

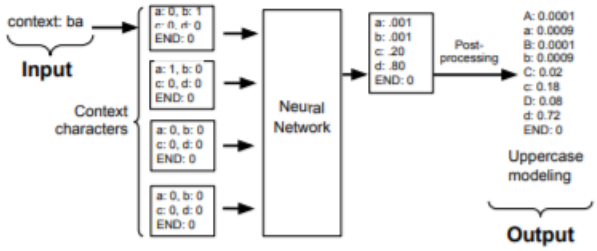


Figure 2.5: Markov Model

Source (Castelluccia, 2012)

2.7.1 Inputs

Figure 2.6 illustrates the construction of password-guessing according to Markov models, special password-ending symbols are relied upon to model the probability of ending a password after a

sequence of characters. For instance, given the word “bad”. The network is being used to predict a ‘d’ given the context ‘ba’ (Castelluccia, 2012).

2.7.2 Processes

The algorithm starts with an empty string, and query the network for the probability of seeing a ‘b’, then seeing an ‘a’ after ‘b’, and then of seeing a ‘d’ after ‘ba’, then of seeing a complete password after ‘bad’ (Ma, 2014). During the learning phase of the model, the prediction process took an input, in this case, ‘b’, and generates, using the internal state, the most likely output. The likely output in this case being ‘a’ followed by ‘d’ to produce the password ‘bad’ (Ma, 2014). The internal state, also known as the memory, of the model is constantly changing throughout the training phase until the model reaches its optimum (Ma, 2014).

2.7.3 Outputs

The probabilities of each next character are the output of the network. Password generation via a neural network model required enumeration of all possible passwords whose probability is above a given weighting or threshold while post processing on the network can infer probabilities of upper or lower-case characters (William, 2016). The process of training neural networks required numerous design decisions, this research took into consideration three different approaches; the context size, the neural network architecture, the training data methodology.

2.8 Existing Solutions

One of the most active research area within the topic of passwords is the study of the quality of human-created secret words under different scenarios, such as password strength meters, different password policies, and the persuasion to introduce random characters into individual password choices (Ma, 2014). Some of the earlier works done under prediction of password include the use of cracking tools, such as John the Ripper (JtR). JtR is one of the most popular tools used for generating password guesses using techniques, such as brute force and dictionary attacks. Its ability to guess password is dependent on the computer’s processing power (Mills, 2021). Another application that is used to perform predictive analysis on passwords is OMEN a faster password guessing tool that uses and ordered Markov enumerator. OMEN was proposed by Narayanan and Shmatikov uses a Markov template based model that assigns probability to letter based segments (Durmuth, 2015). The authors noted that OMEN significantly reduced the speed at with passwords can be guessed over existing platform. This conclusion was derived after OMEN was compared with John the Ripper whereby OMEN successfully guessed more than 40 percent of the passwords

within the initial 90 million guesses (Durmuth, 2015). Durmuth et al. (2015) noted that John the Ripper required eight times as many guesses for it to achieve the same amount of correct password cracks.

2.9 Value Proposition of the Solution

2.9.1 Gaps in Existing Solutions

Although JtR has been used to successfully guess human-created passwords, it has been described by users as being difficult and complicated to use. John the Ripper required an in-depth understanding of the application beforehand hence, limiting the number of users who can leverage its benefits (Wilcox, 2021). While the performance of OMEN against John the ripper were impressive, experimental results performed on OMEN indicated that its approach underperforms significantly compared to variable or higher-order Markov Models (Ma, 2014).

2.9.2 Benefits of Implementing a Neural Network Based Password Predictive, Generation, and Storage Scheme

The benefits of implementing a neural network based password predictive, generation and storage scheme can be derived from the gaps that were identified in existing solution. The scheme sought to provide an appealing, user friendly, and easy to use interface where users can benefit from the application without prior knowledge on how passwords are cracked. Additionally, the use of variable order Markov models ensured that the resulting scheme will perform exceptionally in guessing human-created passwords regardless of the users' computing power. It was noted that none of the existing tools combined password prediction, generation, and storage. In this case, the resulting scheme offered additional benefits to the user by providing a platform where the user can generate complex passwords and securely storing them in the embedded password manager.

2.10 Conceptual Framework

The purpose of this study was to analyse and create an online scheme that can predict, generate and store passwords in real-time. The research focused on human created passwords only. To achieve this goal, this research reviewed a couple of concepts such as the different factors that predispose existing authentication models to attacks. From the literature review, existing gaps were reviewed, highlighting the value proposition of this online scheme. This research focused on incorporating the use of machine learning techniques and derived concepts from the Morkov chain model to develop a scheme that is able to predict human created passwords and generate stronger ones. The scheme relied on the premise that stronger computerized passphrases containing

preferred symbols, special characters and a mix of both upper and lowercase letters, would be generated after the completion of a predictive analysis. Based on the complexity of the resulting passwords, a password manager was used to securely store the passphrases. Figure 2.7 is a pictorial representation of the conceptual framework.

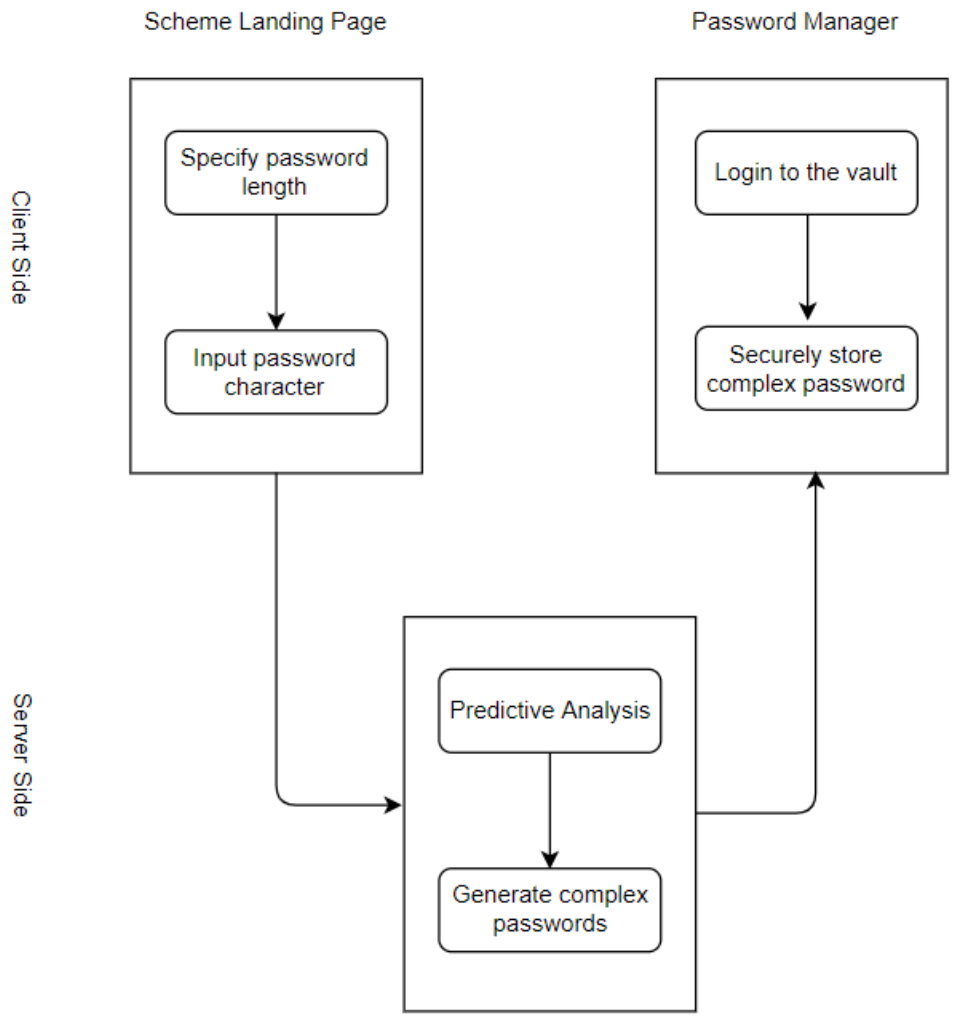


Figure 2.6: Conceptual Framework

2.11 Summary

This chapter was an evaluation of existing publication and other forms of literature on passwords, machine learning techniques, existing solutions, and the value proposition of the proposed online scheme. The concept of password authentication was introduced followed by an in-depth analysis of the various factors that predispose passwords to guessing attacks. Moreover, the chapter uncovered various ways in which machine learning techniques can be used to develop an online

scheme that performs predictive analysis as well as generating stronger passwords based on user preference. The importance of using password managers to securely store complex secret words was also highlighted. This analysis was used to answer the first two objectives of this research. Lastly, chapter two presented a value proposition on the proposed predictive, generative and storage scheme after a thorough comparison with other existing solutions was performed.

Chapter 3: Research Methodology

3.1 Introduction

The literature review highlighted how artificial neural networks can help users build stronger passwords. Therefore, this section revolved around the methodologies that were used to achieve the research goals and objectives. The chapter was a detailed overview of design and methods that were used to create the scheme.

3.2 Research Methodology for Research Objectives 1 and 2

The first two research objective, see chapter 1, subsection *1.3.1 Specific Objectives* were addressed by doing extensive online desk research on the different user authentication approaches and the factors that make passwords vulnerable to guessing attacks. The online desk research involved analysis of existing documents and other online materials, such as reports and publications. The findings of this research were outlined in chapter 2 *Literature Review*.

3.3 Research Methodology for Model Development

Simulation modelling methodology was used to achieve the development and validation of the password predictive and generation scheme. Simulation modelling can be defined as processes involving the development and analysis of a prototype of a perceived physical model in order to predict its behaviour and performance in real-world scenarios over time (Nimbix, 2020).

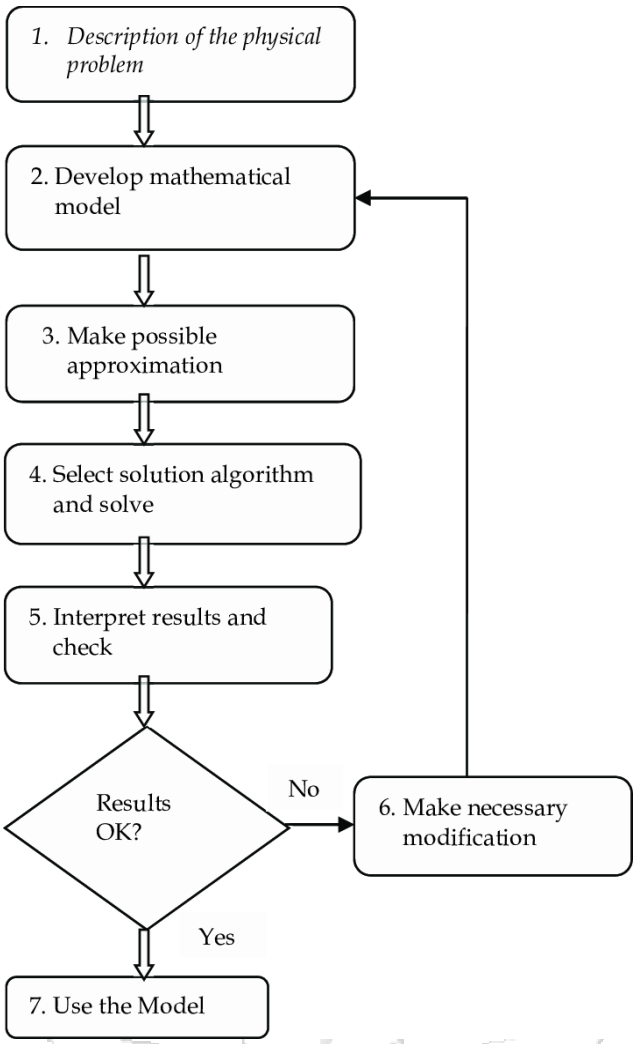


Figure 3.1: Simulation Modelling Process

Source (Perakath, 2006)

The simulation modelling approach was deemed suitable as it involved several techniques which enabled the requirements, and subsequently the scheme to be developed via a series of iterative activities which generally involved the use of prototypes (Nimbix, 2020). Once developed, the prototypes allowed the scheme to be examined and modifications and refinements were made quickly and easily. The prototypes then became the final delivered model.

3.3.1 Description of the Physical Problem

This is the initial phase in simulation modelling that focusses on establishing the purpose of the research study while determining the scope. Typically, simulated models are designed to address certain objectives or to answer questions (Perakath, 2006). This step involved several related activities, such as defining the problem statement and describing the specific objectives which was

done in chapter 1 subsection *1.2* and *1.3.1* respectively. The scope of the study was defined in chapter 1, subsection 1.5 *Scope and Limitations*.

3.3.2 Develop Mathematical Model

The password prediction and generation aspect of this scheme was based on derived concepts of the Markov chain model. Developed by a Russian mathematician, Markov chain model is a mathematical model of stochastic processes used in text generation to predict the probability of the next letter based on the state of the previous one (Adyatama, 2020). In the case of password prediction and generation, Σ was used to denote all the alphabetic characters that can be used in creation of passwords. This premise followed an assumption that all passwords would be of lengths between *L* and *U*, with *L* and *U* denoted by some value. Σ was set to include all the printable American Standard Code for Information Interchange (ASCII) characters which are 95 in total while the values of *L* and *U* were set at 6 and 20 respectively. Based on this mathematical model, the accepted password would be:

$$\Gamma = \bigcup_{\ell=L}^{U} \Sigma^{\ell}$$

3.3.3 Make Possible Approximation

The third step in simulation modelling involves some approximations that are used to facilitate the solution. According to Ali (2012), there is an interrelation between steps 2 and 3 which are done iteratively. The main assumption that was made in the conceptualisation of this scheme was that, a password model denoted as *p* is considered to be complete if and only if the resulting string in Γ is assigned a non-zero probability. In this case, for *p* to be useful in password prediction, it should have enumerated efficiently. For instance, any integer *S*, will execute proportionally in time to *O* ($S \cdot |\Sigma| \cdot U$) in order to output the *S* passwords which contain the highest probabilities according to the model *p*. Hence, p: $\Gamma \rightarrow [0,1]$.

3.3.4 Method of Solution

At this step, the type of data to be used in the development and training of the model is determined. In this case, this research employed a dataset collected from GitHub, Top204Thousand-WPA-probable-v2.txt file containing a total of two hundred and four thousand unique passwords (Berzerk0, 2018). The quantity of the dataset provided in this text file was deemed enough to train and test the developed machine learning model.

3.3.5 Model Translation

The fifth step involved the translation of the model equation and approximation into the programming language. In this case, the selections may range between simulation programming and general purpose languages. This study used python programming language in the development of the prediction and generation model as discussed in chapter 1, subsection 1.5 *Scope and Limitations.*

3.3.6 Validation

Model validation refers to the process of determining whether or not the theories, mathematical equations and assumptions which were used to define the conceptual framework are correct. Validation also ensures that the relationship between the problem, structure, and logic correspond to the intended purpose of the model (Sargent, 2011). Cross-validation was the technique used during the validation of the prediction and generation model. This technique is used to evaluate a model by testing its accuracy or performance (Joby, 2021). During cross-validation, a subset of the dataset is reserved against training. The model is then tested using this subset in order to evaluate its efficacy levels.

3.3.7 Deployment

The final step in simulation modelling is model deployment, a process used to make models available in production environment. Deployment is a crucial step as it provides a platform whereby users are able to interact with the model by issuing commands or input and expecting an output (Baier, 2019). The deployment of the model was done using flask, a python based micro web framework (Pallets, 2010).

3.4 Research Methodology for Password Manager Development

Rapid Application development (RAD) methodology was used to design, develop and test the password manager aspect of the scheme and to validate its effectiveness in securely storing generated complex passwords. RAD is a development lifecycle designed to give faster development and high-quality results (Singh, 2019).

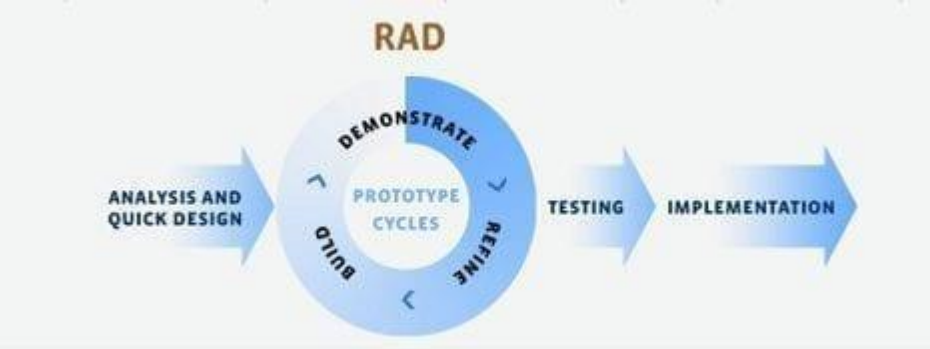


Figure 3.2: Rapid Application Development

Source (Singh, 2019)

3.4.1 Requirements Phase

The initial phase involves collection and thorough analysis of functional and non-functional requirements. Proper understanding of these requirements is necessary in ensuring that the final product meets the expectation (Pedamkar, 2020). The password manager's functional and non-functional requirements are discussed in chapter 4, subsection 4.2.1 and 4.2.2 respectively.

3.4.2 Design Phase

After the requirements are defined, the design phase commenced. In this phase, a detailed analysis of the various activities within the system and how they relate was used to inform the structural composition of the password manager. According to Pedamkar (2020), system designs should be flexible enough in order to accommodate addition of new components. The design phase involved the creation of data flow diagrams, such as use cases and sequence diagram.

3.4.3 Development Phase

This phase involves the development of the password manager. The development process is iterative to allow integration of all functional and non-functional requirements. The tools used during the development phase include:

- VueJs and Bootstrap were used in the development of the front-end which allowed effective user interaction with the system.
- PHP Laravel Framework was used in the development of the back-end.

3.5 Testing

Once the development phase was complete, the password manager was embedded into the predictive and generation model. They were both tested as one entity to ensure that the prediction, generation and storage functionalities worked as expected. The following tests were carried out;

- Functional test
- Usability test
- Compatibility test

3.5.1 Functional test

The requirements, both functional and non-functional, were tested in line with the model algorithm to realize its functions. This test was conducted via distribution of a survey questionnaire form to 10 different participants. A sample questionnaire was attached in the appendix page with resulting answers. The model performed as expected by being able to predict a password given a character and length, generate and suggest a stronger password using various combinations based on user preference.

3.5.2 Usability Test

Usability testing was conducted to ensure that the application met the required aesthetic values. This was an important test since system users always like applications that are appealing to the eye. The resulting findings were compiled in chapter 5, subsection *5.5.2: Usability Test*.

3.5.3 Compatibility Test

The scheme was tested against different browsers, the compatibility to different browsers was an indication that the scheme can be integrated into other existing solutions. The browsers included:

- Safari - OS X.
- Google Chrome

3.6 Implementation

The aim of this stage was to demonstrate that the proposed scheme satisfied the specified requirements and objectives. Several iterations were done during the system design and development phases before the cutover phase.

3.7 Ethical Considerations

This study was guided by the frameworks of scientific methods and procedures. All the sources of information used in this research are credited. Personal information obtained, such as passwords were not leaked to other parties.

3.8 Summary

This chapter highlighted the methodologies that were used in undertaking this research. The first two research objectives were addressed using extensive desktop research and the findings documented in chapter 2 *Literature Review*. Simulation modelling methodology was used to address the design, development and validation of the password prediction and generation model while rapid application development methodology highlighted how the password manager was developed. After completion of development, the password manager was embedded into the model scheme whereby testing commenced. This chapter also highlighted the ethical consideration of the research.

Chapter 4: System Design and Architecture

4.1 Introduction

This chapter described the procedure in which the algorithm scheme was designed and optimized for guessing passwords given a few parameters such as the length of the password. The chapter also highlighted the scheme's design and architecture. A detailed description of the scheme structure was given to deepen the understanding of the authentication process. Data flow within the various system modules and their triggers was also laid out. A greater part of the design and architecture of the proposed scheme was informed by findings from the online desk research conducted throughout this study. Based on the findings, the researcher was able to incorporate the user requirements elicited from the survey.

4.2 Requirements Analysis

Requirements analysis stage involved gathering necessary information from stakeholders that informed the development of the model. From the information gathered, functional and non-functional requirements were distinguished. The functional and non-functional requirements were then used during scheme design and development.

4.2.1 Functional Requirements

To have a successful model, the requirements must be made clear and concise. Some of the functional requirements that were deemed important in this research include

- The resulting scheme should allow user input
- The model should be able to predict user passwords in real time
- The system should allow users to input various parameters, such as their preferred password length, symbols, and characters.
- The model should generate a stronger password while implementing the same user parameters
- The scheme should not store user information

4.2.2 Non-functional Requirements

Non-functional requirements that informed this study were divided into two, hardware and software requirements;

Hardware Requirements

i. Machine: Intel Core i3 or Higher
ii. Clock speed and processor: 2.5 GHz or higher

iii. System Memory: 4GB or higher

Software Requirements

i. Operating system: Windows 10/8/7
ii. Python 3.7.*
iii. Vuejs 2.6.*
iv. Bootstrap 5.0.*
v. Laravel framework

4.3 Scheme Architecture

System architecture is a conceptual model that was used to outline the structural design of the scheme. This conceptual model was instrumental in addressing how various system component interact to achieve the system functionality. This section provides an understanding of how the scheme design, structure and user requirements must be supported by the application. The scheme's architecture in figure 4.1 was based on the concepts of Markov models for measuring password strength. Similar to Markov models, neural networks in this model were trained to generate the next character of a password given the preceding characters.

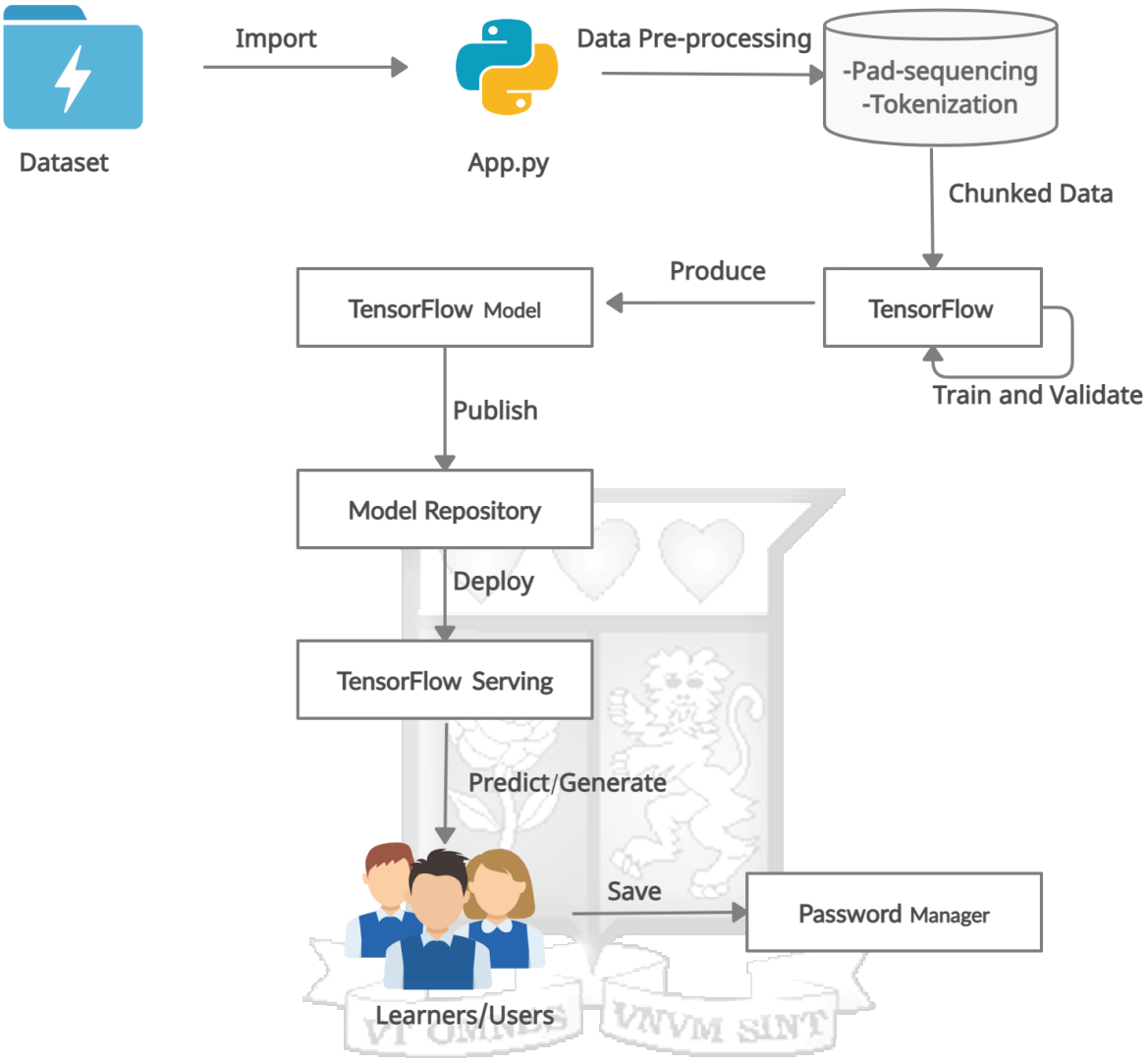


Figure 4.1: Scheme Architecture

4.4 Design Tools

4.4.1 Use Case Diagram

The interaction of the users of the system with the various functionalities of the scheme is illustrated in the use case diagram in Figure 4.2. This use case highlights all the major

functionalities that the scheme offers. All the external systems and users interacting with the proposed solution have also been highlighted.

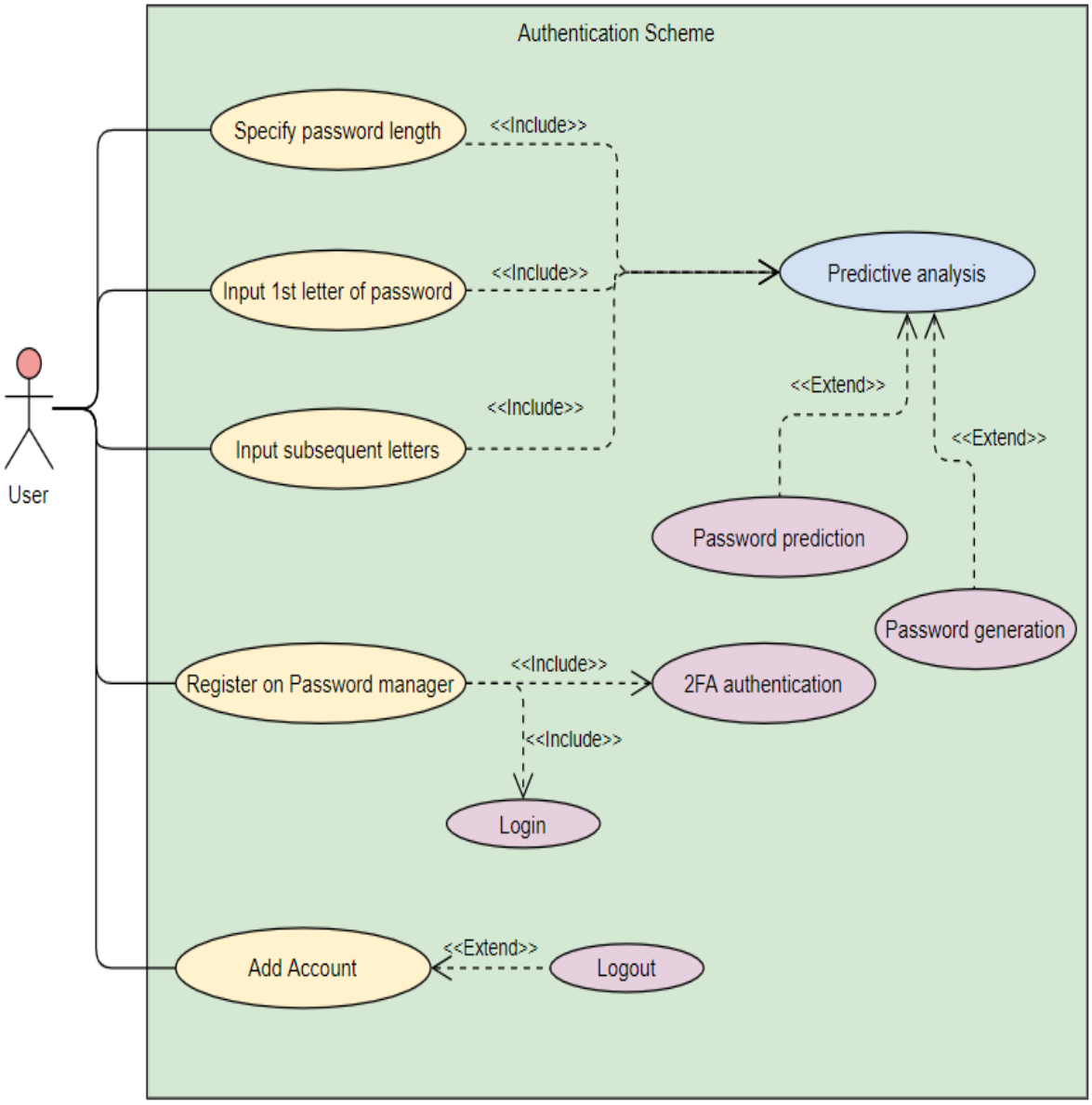


Figure 4.2: Use Case

4.4.1.1 Use Case Description

Tables 4.1, 4.2, and 4.3 are detailed description of the processes involved in predictive analysis, password generation and storage within a password manager, respectively.

<table>
<tr><th colspan="2">Password Prediction Use Case Description</th></tr>
<tr><td colspan="2">Scheme: An Online Neural Network Based Password Prediction, Generation and Storage Scheme</td></tr>
<tr><td colspan="2">Use Case Name: Password Prediction</td></tr>
<tr><td colspan="2">Primary Actor: Young Learners/User</td></tr>
<tr><td colspan="2">Goal: Predict subsequent password content given an initial input.</td></tr>
<tr><td colspan="2">Relationships
Includes
Extends</td></tr>
<tr><th colspan="2">Normal/Basic Flow of Events</th></tr>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>2. User inputs their preferred password length and starts typing it in in the available password text field</td><td>1.Scheme provides a password input text field and a preferred length specification

3. Scheme begins the prediction process which changes with every added character and stops when the whole password has been predicted.</td></tr>
<tr><td colspan="2">Expected Outcome: Prediction ends before the user finishes writing their password</td></tr>
</table>

Table 4.1: Password Prediction Use Case Description

<table>
<tr><th colspan="2">Password Generation Use Case Description</th></tr>
<tr><td colspan="2">Scheme: An Online Neural Network Based Password Prediction, Generation and Storage Scheme</td></tr>
<tr><td colspan="2">Use Case Name: Password Generation</td></tr>
<tr><td colspan="2">Primary Actor: Young Learners</td></tr>
<tr><td colspan="2">Goal: Generate a passphrase that contains characteristics of a complex password</td></tr>
<tr><td colspan="2">Relationships
Extends</td></tr>
<tr><th colspan="2">Normal/Basic Flow of Events</th></tr>
<tr><th>Actor</th><th>System</th></tr>
</table>

2. User specifies the different characters that should be contained in the generated password	1.Scheme provides an option for the user to generate a password containing characters considered elements of a stronger password 3. Scheme generates and displays a password containing user specifications.
Expected Outcome: A complex password is generated	

Table 4.2: Password Generation Use Case Description

Password Manager Use Case Description	
Scheme: An Online Neural Network Based Password Prediction, Generation and Storage Scheme	
Use Case Name: Password Manager	
Primary Actor: Young Learners/User	
Goal: The generated password is stored in a password manager.	
Relationships • Extends	
Normal/Basic Flow of Events	
Actor	**System**
1. First time user is required to register 3. Verify email using the link sent to their emails 4. Login to the password manager 5. Click on add account 6. Add the generated password for secure storage 7. Logout	2. Two-factor authentication
Expected Outcome: Generated password is securely stored	

Table 4.3: Password Manager Use Case Description

4.4.2 Sequence Diagram

The main features within the authentication platform include performing a password prediction analysis, generating a stronger passphrase given user specifications, and encrypting and storing the generated password in a password manager. Figure 4. 3 shows the major flow of events and data therein for major functionalities of the proposed scheme.

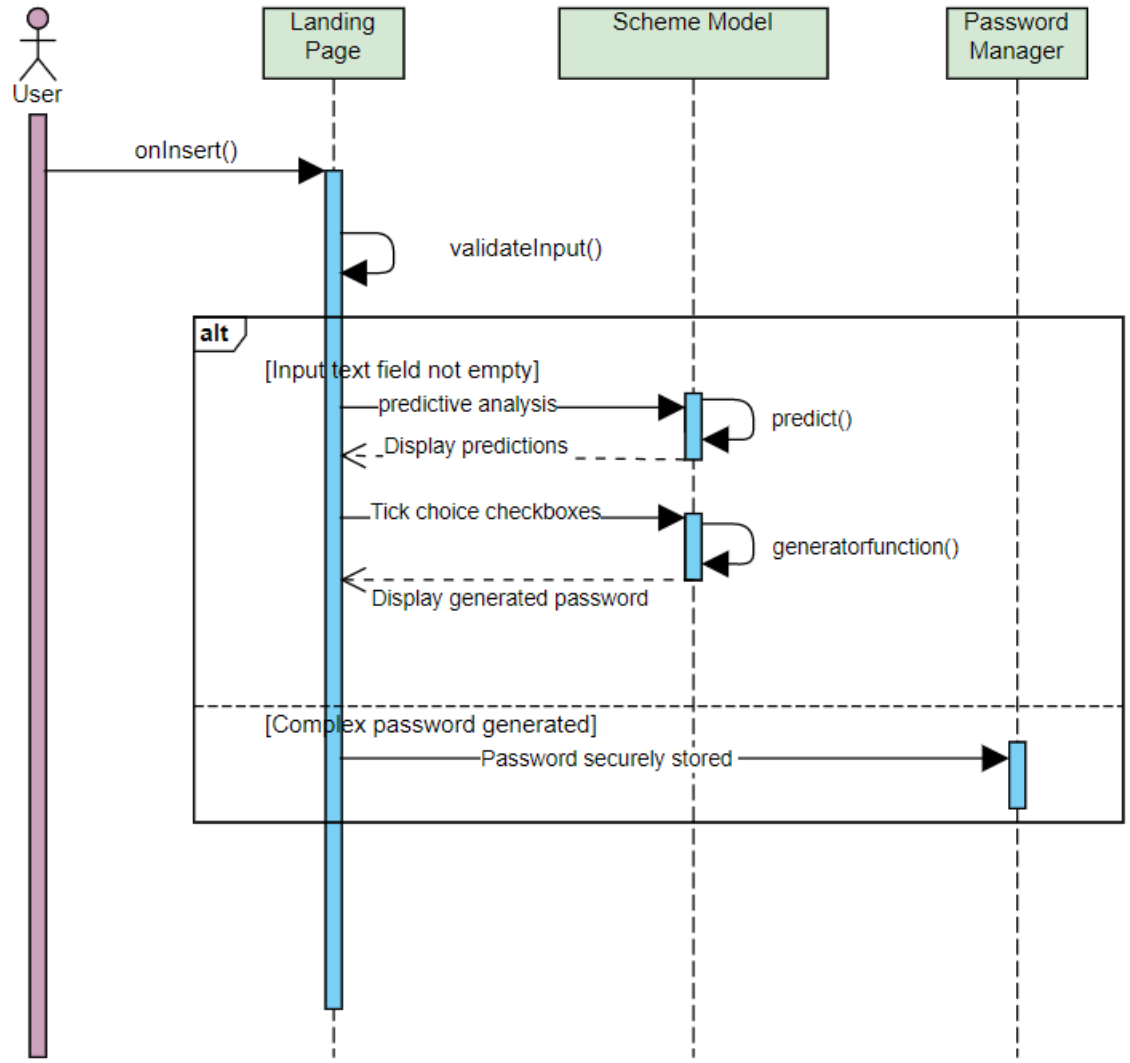


Figure 4.3: Sequence Diagram

Input Validation

Using the function, validateinput(), the scheme was able to trigger the prediction process only when the user inputs text, symbols or number characters into the password text field. Subsequent

processed depended on the validation of user input. The function was used to check for two variables, input type and length.

Prediction Function

The function predict() was triggered after the user's input was validated. In this case, this function works with the training dataset making use of the learned values with then maps and predicts subsequent labels within the testing dataset.

Generator Function

The generator function used in this scheme is similar to a function that returns values in an array. In this case, this function contains parameters which when called, generates a sequence of randomised characters.

4.4.3 Flow Chart

Prediction was based on two data inputs, that is, the users and the scheme's training dataset. User input was compared against the datasets for ease of prediction. The process stops when prediction is complete. Figure 4.4

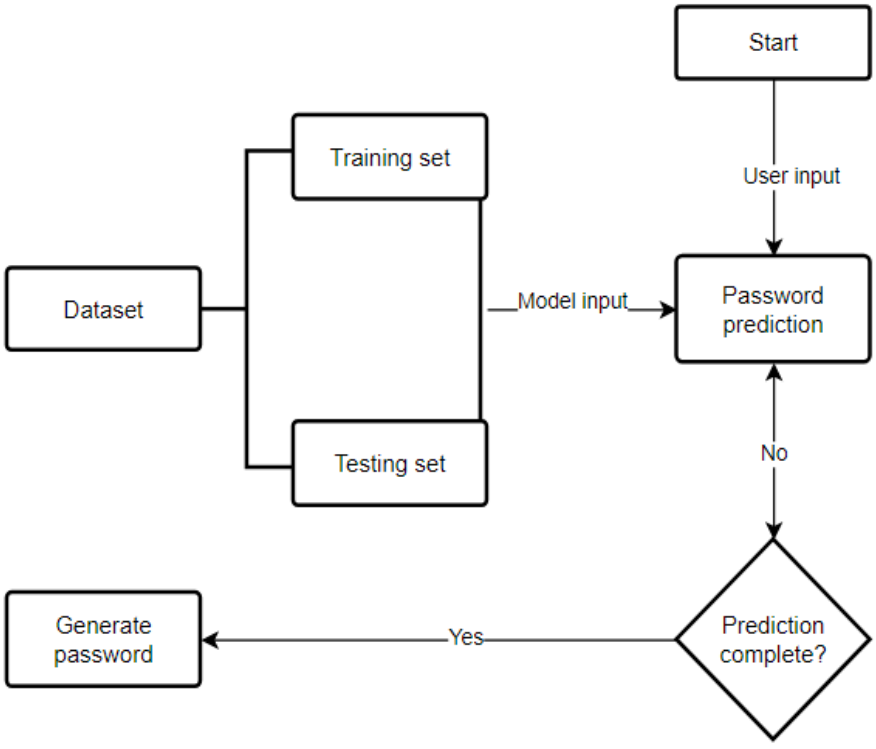


Figure 4.4: Flow Chart

4.5 Summary

This chapter reviewed the functional and non-functional requirements necessary for the development of the password prediction scheme. The system architecture was described in detail as well as the various design tools including use cases, sequence diagram, and data flow diagram. The various use cases were also outlines and discussed.

Chapter 5: System Implementation and Testing

5.1 Introduction

To display the scheme's functionality practically, first time users were given the option of testing their password strength by how fast the scheme could predict it. As the user inputs their password, the model begun to predict the final outcome which changes depending on what the user types in. The scheme then generated a more secure password which was used to login. Since a computer generated passphrase is normally complex and hard to memorise, the scheme relied on an embedded password manager to encrypt and securely store the new password.

5.2 Implementation Environment

5.2.1 Hardware Requirements

The scheme development was done on a personal computer (PC) installed with Windows 10 operating system, the specifications can be seen in the table 5.1

Device Name	DESKTOP-0NKO3QB
Processor	AMD A12-9700P RADEON R7, 10 COMPUTE CORES 4C+6G 2.50 GHz
RAM	8.00 GB (7.47 GB usable)
Device ID	B1F86FAB-319B-4466-ACA8-707EA9B75393
Product ID	00325-96037-27225-AAOEM
System Type	64-bit operating system, x64-based processor
Additional support	Pen and touch support with 10 touch points

Table 5.1: Hardware Specifications

5.2.2 Software Requirements

The Anaconda Distribution version 2019.03 (Anaconda, 2020) was used to install Python 3.7.3. Anaconda is a high performance distribution that pre-installs several relevant Python packages allowing easy installation and management of packages, dependencies and environments.

Some of the important packages that were used in the algorithm development are:

1. ***Keras version 2.2.5***– for building the neural network (Keras, 2021).

2. ***TensorFlow version 2.4***– a deep learning library used to develop and train machine learning models (TensorFlow, 2020).
3. ***NumPy version 1.20*** – for numerical computations in the algorithm (NumPy, 2020).
4. ***Pandas version 1.2.4***– a data manipulation library that was used to load the data (Pandas, 2020).
5. ***Vuejs version 2.6-*** a progressive framework that is used to build user interfaces (Vue.js, 2021).
6. ***Bootstrap version 5.1.0-*** a free, open source framework containing Cascading Style Sheets (CSS) used in front-end development (Bootstrap, 2021).
7. ***Laravel framework version 8.x-*** a PHP web application framework (Laravel, 2021).

It is important to note that not all Keras packages were imported in their entirety. Figure 5.1 displays the modules imported from the keras library for the development of the prediction and generation model.

```
# keras module for building LSTM
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Embedding, LSTM, Dense, Dropout
from keras.preprocessing.text import Tokenizer
from keras.callbacks import EarlyStopping
from keras.models import Sequential
import keras.utils as ku
```

Figure 5.1: *Imported Keras Modules*

5.3 Model Development

5.3.1 Loading the Dataset

After all the necessary packages were installed, the next step involved uploading the dataset text file into the python environment using the Pandas library. Berzerk0 (2018) had cleaned up the data contained in the Top204Thousand-WPA-probable-v2.txt file by removing any internal duplicates and ensured that all the passwords used the same newline characters. The dataset was comprehensive as it contained data collected from other GitHub directories, such as SecList (Miessler, 2019) and websites, such as Weakpass (Netmux, 2020). While compiling this file, the author considered the number of times a password was found across all the files to be an approximation of its overall popularity, for instance, if a password was found in less than 5 files, then it was considered to be less common, on the other hand, if a password was found in more than 350 files, then it was considered to be incredibly popular. In the final compiled list, the most

common or frequently used passwords were placed at the top while entriesthat were considered non-common appeared at the bottom of the list. The figure 5.2 shows a sample of the first 10 passwords contained in the dataset.

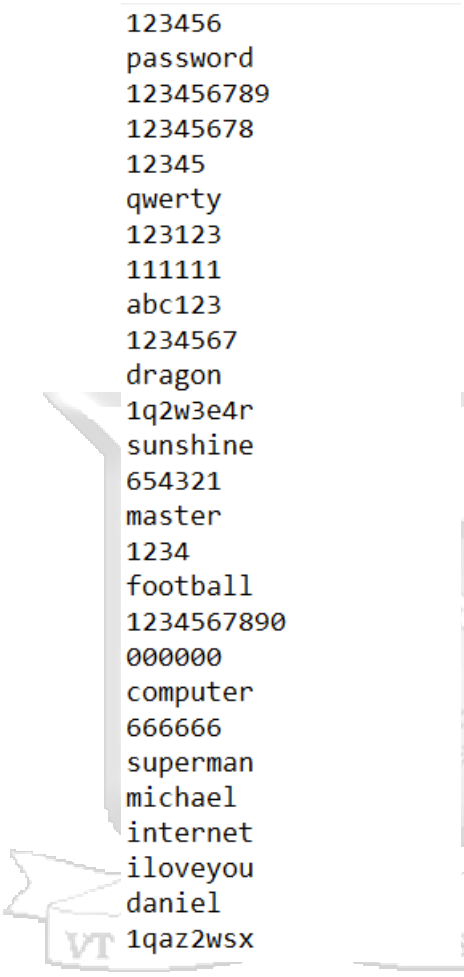


Figure 5.2: *Sample Data*

5.3.2 Data Pre-Processing

Generating Sequence of Tokens

The raw data fed into this scheme consisted of continuous text, referred to as the Corpus, which was first split into words. These words are what is referred to as tokens, keras has an inbuilt function for tokenisation, get_sequence_of_tokens () see figure 5.3. Once the tokenizer was fit on the dataset, it was used to encode data in the train and test datasets.

```
def get_sequence_of_tokens(corpus):
    ## tokenization
    total_words = len(letters_dict) + 1

    ## convert data to sequence of tokens
    input_letter_sequences = []
    for word in corpus:
        ww = [letters_dict[j] for j in word if j in letters_dict]
    #     print(ww)

        for i in range(1, len(ww)):
            n_gram_letter_sequence = ww[:i+1]
            input_letter_sequences.append(n_gram_letter_sequence)
    return input_letter_sequences, total_words

input_letter_sequences, total_words = get_sequence_of_tokens(corpus)
input_letter_sequences[:10]
```

Figure 5.3: *Tokenisation*

Sequence Padding

After tokenisation and data containing sequences of tokens was generated, there was the possibility of different sequences having different lengths. To make their lengths equal, sequence padding was introduced by the use of the *pad_sequence()* function of keras. See figure 5.4

```
def generate_padded_sequences(input_sequences):
    max_sequence_len = max([len(x) for x in input_sequences])
    input_sequences = np.array(pad_sequences(input_sequences, maxlen=max_sequenc
e_len, padding='pre'))

    predictors, label = input_sequences[:,:-1],input_sequences[:,-1]
    label = ku.to_categorical(label, num_classes=total_words)
    return predictors, label, max_sequence_len

predictors, label, max_sequence_len = generate_padded_sequences(input_letter_seq
uences)
```

Figure 5.4: *Sequence Padding*

5.3.3 Train-Test Split

Here the Scikit-learn library was used to split the data and separate the training set from the testing set. This was done by specifying the size of the test data to be 20%. The remaining 80% was used as the training data. After the above data preparation stages were completed, the process of creating the model commenced.

5.3.4 Create Model

The first step in model creation involved defining the sequential model in the order that the neural network layers were stacked up, this also represents how the layers are connected. The model consisted of three layers, that is embedding, LSTM, and a Dense layer as depicted in figure 5.5.

In-between these layers, Dropouts were also added with the purpose of avoiding overfitting the model during training.

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 25, 10)	830
lstm_1 (LSTM)	(None, 100)	44400
dropout_1 (Dropout)	(None, 100)	0
dense_1 (Dense)	(None, 83)	8383

Figure 5.5: *Sequential Layers*

5.3.5 Configure Model

The process of model configuration begins once the model layers have been created. In this case, the *compile()* function is used to configure the metrics, loss type and optimiser. Although numerous optimisation functions exist, the Stochastic Gradient Descent (SGD) was preferred over the rest due to its fast computational capabilities. During optimisation, the loss function was calculated by evaluating the scores after every training iteration. In this case, losses are used to determine the quality of data computed by the model (Pedamkar, 2020).

5.3.6 Model Training

Model training was conducted using the *fit()* function whereby the training dataset was converted into a metric of inputs *x* and an array of outputs *y*. The scheme was trained using 50 iterations, epochs, the number of iterations can be increased or decreased depending on the type and quantity of the training dataset. The resulting weights after training were also recorded. A decreasing loss value after every iteration was noted which was an indication that the training dataset was not under fitted.

5.3.7 Model Validation

The process of cross-validation was used to validate that the password prediction and generation scheme had the capacity to correctly guess user passwords in real-time and generate stronger complex passwords after the predictive analysis was complete. After the model had been trained using the training dataset, validation commenced by testing the performance of the model using unseen data, that is, test dataset. Figure 5.6 depicts the gradual decrease in losses as the model was

being trained. This decrease was an indication that the model was learning new information with each iteration.

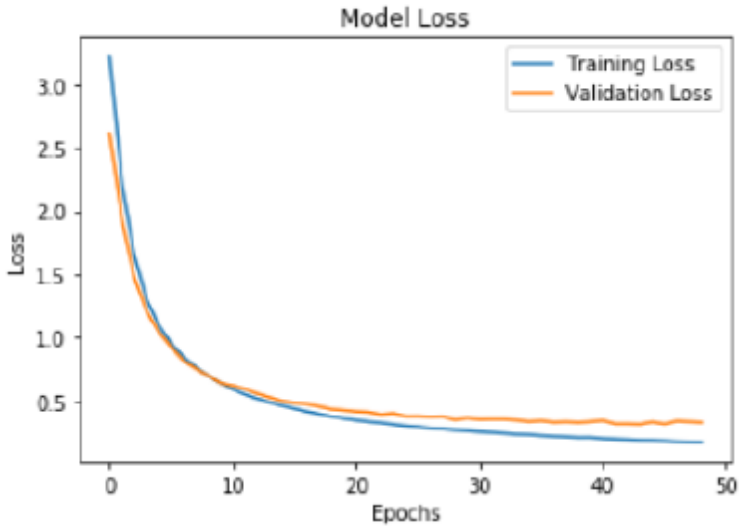


Figure 5.6: *Decreasing Loss value*

Another graph that is used to explain validation is model accuracy. Figure 5.7 showed a gradual increase in model accuracy with every training iteration. From the figure, the model registered an accuracy rate of 0.93 which is equivalent to 93% and a training loss of 0.3.

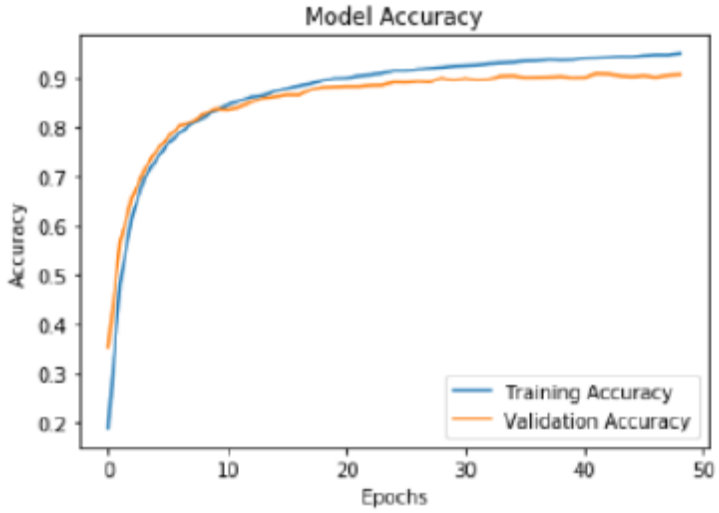


Figure 5.7: Scheme Accuracy

5.3.8 Model Deployment

After evaluating the model accuracy, the process of deployment commenced using Flask, a web framework written in python. Deployment was crucial because it provided an interface whereby

Internet users can interact with the model. A couple of decorators were used during the deployment process. The first one was @app.route ('/index') which maps the *index()* function and the method to return the main Hypertext Mark-up Language (HTML) page titled index.html. Another decorator is the @app.route ('/predict') which functions by mapping the predict() method. This method functions by taking the user input, performing pre-processing of the data and outputting the possible guesses depending on the next character probabilities.

5.4 Password Manager Development

5.4.1 Back-End

The password manager's back-end was developed using the Laravel framework due to its simplicity, clarity and end-user oriented (Barnes, 2021). the first step involved the installation of the Laravel framework using the command 'composer install' followed by a database set-up. MYSQL was the preferred database which contained three tables, users', account, and password reset table. The users' table was used to store user information during signup to facilitate easier subsequent logins while the accounts table was used to save complex password that were generated from the model. The model-view-controller architecture played a crucial role in facilitating the input, processes, and output functions between the back-end and front-end.

5.4.2 Front-End

The front-end portion of the password manager was developed using Vuejs and bootstrap. Vuejs was preferred over other JavaScript libraries because of its fast development time (Antala, 2020). On the other hand, bootstrap provided appealing styling elements, supports the latest browser versions, and provides a means in which web processes share the same design patterns (Colin, 2018). Once the password manager was complete, it was embedded into the model's architecture so that users can access all three functionalities, that is, prediction, generation and storage within one platform.

5.5 Client Side User Interface

The client side of the application runs on a web platform, allowing users to access the scheme on any Internet enabled device. The main use of the client side is to provide a platform whereby young learners can interact with the scheme to realise its functionalities. The development of the client side was done using Django, an open source python-based web framework that implements the model-template-view architecture. The client-side of the scheme contains three major functionalities which include:

i. Password prediction
ii. Password generation
iii. Password storage

5.5.1 Password Prediction, Generation, and Storage

Once the scheme was loaded onto the users' browser, a single page website was displayed with different sections, the first one being password prediction followed by the password generation segment. The user will be prompted to specify their preferred password length, followed by a textbox that the user will use to input their choice passwords. The scheme will then perform a predictive analysis as the user continues to input their password as can be viewed in figure 5.8. The analysis acts as a password meter such that the strength of the password will depend on the ability of the scheme to correctly predict the users' password. In this case, fast prediction will indicate that the password is weak while incorrect prediction will determine that the users' password is strong and can withstand different hacking techniques. Similar to Markov model, the scheme begins its prediction with an empty string which is then queried based on the probability of the first and subsequent letters.

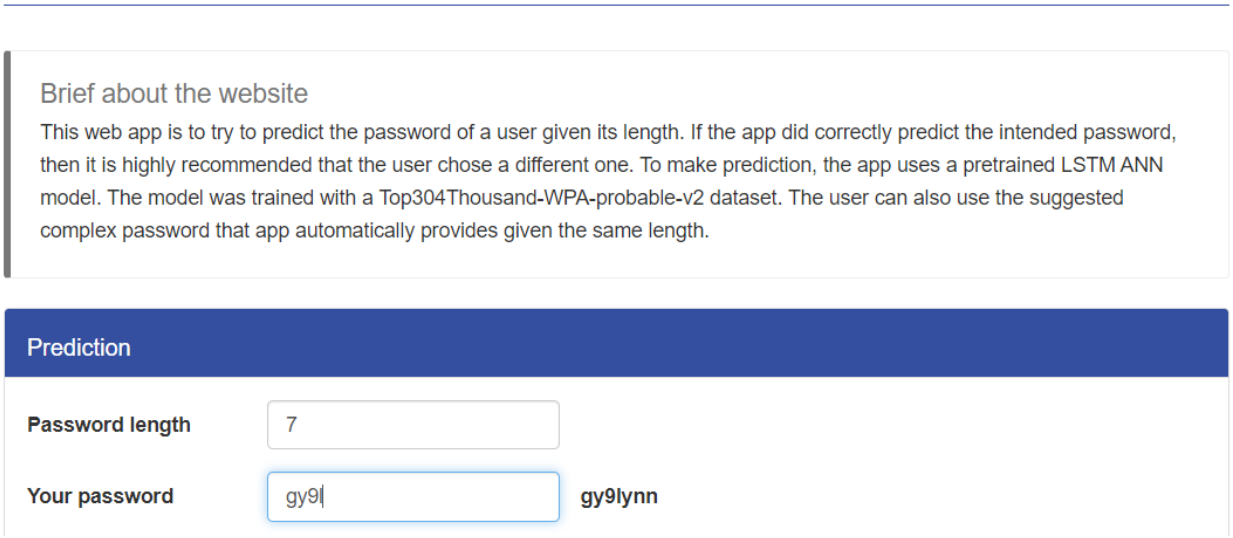


Figure 5.8: Password Prediction

Following the completion of the predictive analysis, the user will have the opportunity to either use a model suggested password or generate a complex password by ticking either of the two checkboxes provided while maintaining the original preferred length. Figure 5.9 displays the process of password generation.

Figure 5.9: Password Generation

Due to the complexity of the generated password, the challenge of ease to remember might arise. In this case, the user is encouraged to store the generated password within the schemes password manager. This manager can be accessed by clicking the button located at the bottom right corner of the user interface. This action will redirect the user into a login page, as can be viewed on figure 5.10, where they will have to authenticate themselves prior to accessing the contents of the password vault.

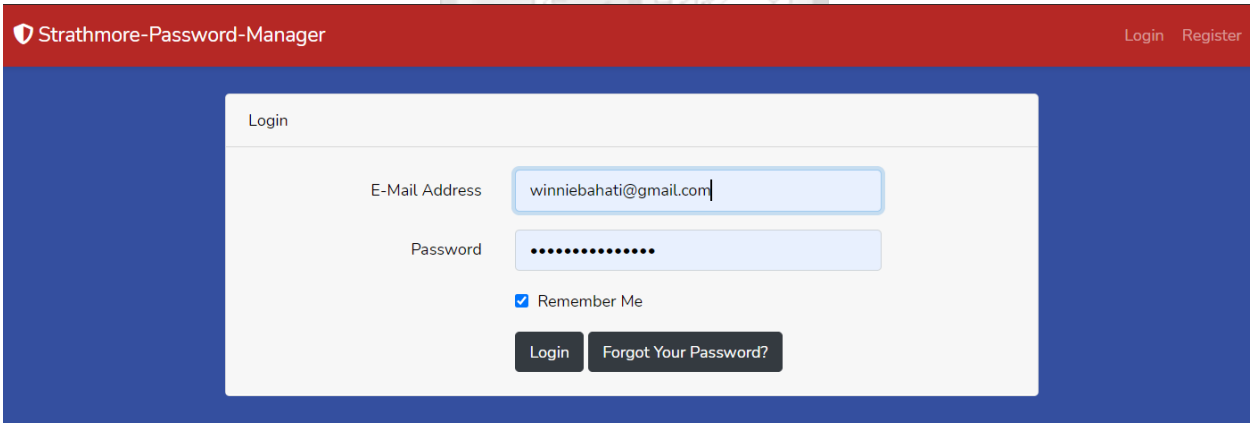


Figure 5.10: Password Manager Login Page

For first time users of the password manager, they will be required to register using a registration form similar to the one displayed on figure 5.11. Two factor authentication protocol was implemented during registration whereby user will be required to verify their email addresses. Once verification is complete, the user can proceed to login into the password manager.

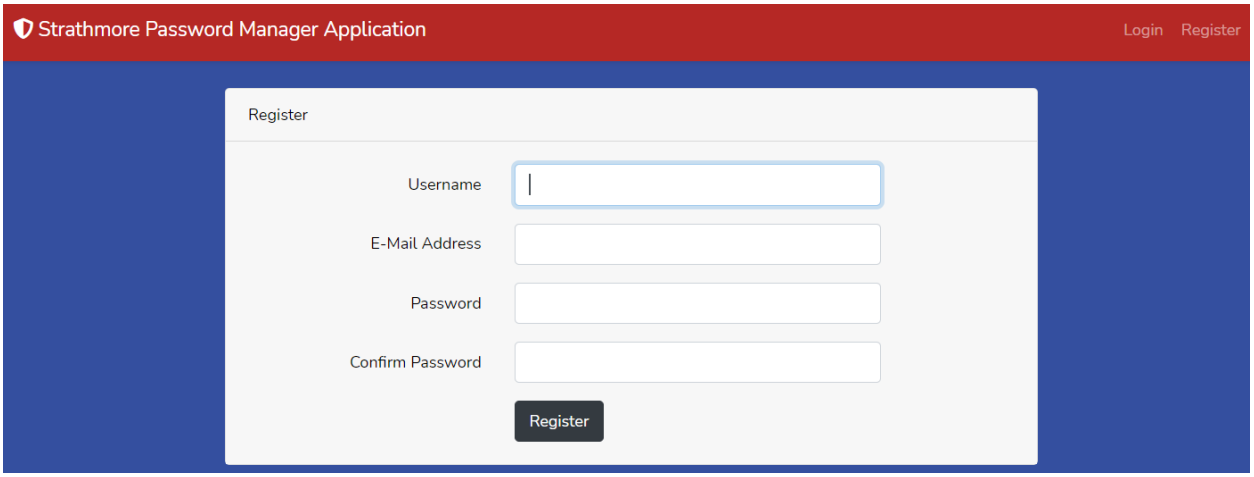


Figure 5.11: Password Manager Registration Page

After successfully login, the user will be able to add and store their generated passwords by clicking the "add account" tab as can be seen on figure 5.12.

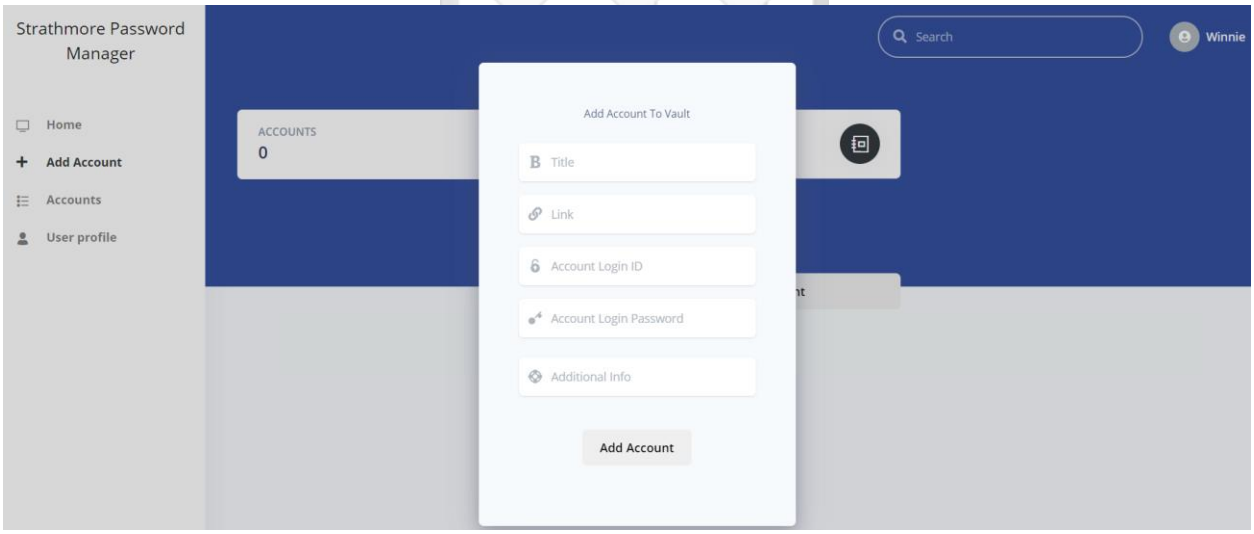


Figure 5.12: Password Manager Landing Page

5.6 Scheme Testing

Testing was a critical part of the scheme's lifecycle because it aided the researcher in identifying whether or not the research objectives were achieved. Through testing, the researcher was also able to identify what improvements would be needed in subsequent versions of the system. Lastly, testing was instrumental in verifying the functionality of the various modules and their inter-functionality thus confirming whether or not the entire scheme worked seamlessly. Testing was carried out in three broad areas: functionality, usability, and browser compatibility.

5.6.1 Functional Test

A thorough functionality test was conducted via a survey containing six scheme related questions. The survey was developed on Google forms and distributed via email to a total of 10 people. The sample size was deemed appropriate to demonstrate the functionality of the scheme. Out of the 10 participants, 3 members stated that safari was their preferred browser while the rest used Google Chrome. Everyone reported that the scheme was functioning according to its requirements. A sample of responses is available in the Appendix page. Some of the data collected during the survey was used to conduct a robustness test. Table 5.2 details the different demographics of the participants of the survey questionnaire, while Tables 5.3 is a detailed test cases that sought to test the schemes functionality from the point of view of the user.

Personal Attribute	**Response**	**Total**
Gender	Male	4
	Female	6
Age	15-20	3
	21-24	5
	25-28	1
	Above 28	1
Occupation	Student	8
	Employees	2
Have you used an online scheme that performs predictive analysis?	Yes	0
	No	10
Have you used an online scheme that generates passwords based on user specifications?	Yes	0
	No	10

Table 5.2: Participant Demographics

Item	**Value**
Objectives	- To identify items to be tested - To detail the test steps

	- To note expected outcomes - To define the test environment
Test Items	- Functionality Testing - Usability Testing
Test Features	- Password prediction - Password generation - Password storage

Test Steps

Pre-Condition:

The user must have the latest version of their preferred browser installed on their personal computers. Furthermore, the computers should be connected to the Internet.

Post-Condition:

When the user inputs the uniform resource locator (URL) into their browser, they should be redirected to the user scheme's landing page

Step	Action	Expected Response	Pass/Fail	Comments
1	Start application by inputting its web address into the web browser, then hit enter to load the application.	Home page screen is now visible to the user	Pass	None
2	The user is required to specify their preferred password length and input the first character of their password.	Application begins its predictive analysis	Pass	None
3	The user is given the option of generating a complex password	The user is not restricted by the number of	Pass	None

	by choosing its components by checking the various checkboxes available	checkboxes they can choose from		
4	A password containing complex characteristics is generated	The generated password contains user specifications	Pass	None
5	The user logs in to the password manager	Successful login	Pass	None
6	Upon successful login, the user adds the password to be store by clicking on the "Add account" tab	The password is added and securely stored	Pass	None

Table 5.3: Functionality Test Case

5.6.2 Usability Test

Usability testing was conducted to ensure that the application met the required aesthetic values. This was an important test since system users always like applications that are appealing to the eye. Table 5.4 describes the tests that were done to ensure the application developed from this study met the intended visual appeal.

Test Case Name: Scheme Usability **Date Tested**: 24th March, 2021 **Tested by**: Winnie Bahati **Test Description**: Step by step usability test
Test Steps
Pre-Condition:
Post-Condition:

Step	Action	Expected Response	Pass/Fail	Comments
1	User can see and read all the instructions	Instructions are clearly visible	Pass	None
2	User can press/click to select where necessary	All the necessary menu items are clickable	Pass	None
3	Overall appearance and colour mix and balance	Colours were well mixed and balanced	Pass	None

Table 5.4: Usability Test Case

5.6.3 Browser Compatibility Test

The compatibility test was conducted on two browsers, Safari and Google Chrome, that were installed on three of the most popularly used operating systems including Windows, MacOS, and linux. The resulting findings were documented in table 5.5 .

Browser	Compatible
Google Chrome	Yes
Safari	Yes

Table 5.5: Browser Compatibility Test

5.7 Summary

This chapter focused on the implementation process of the scheme, highlighting how various processes, such as tokenisation and sequence padding were performed. The screenshots of the user interface were also discussed in details. After implementation, the scheme was validated using four distinct tests, functionality, usability, accuracy, and browser compatibility. The functionality test was used to inform whether or not the scheme was in line with the stipulated functional and non-functional requirements, the second test measured the aesthetic value of the scheme followed by an accuracy measure that resulted into a 91% correctness with regards to the schemes' ability to predict passwords. The last test confirmed that the scheme can operate on both safari and google chrome browsers that had been installed on three of the popularly used operating systems.

Chapter 6: Discussion

6.1 Introduction

This chapter outlined the various methods used to achieve the different research objectives highlighted in chapter 1. An in-depth discussion of each objective was conducted followed by highlighting the advantage and limitations of the study.

6.2 Research Objectives Discussion

6.2.1 Objective 1

Extensive desk research was conducted to identify the character composition of password, factors that may predispose human-created passwords to attacks, and the role of password managers in storing passwords. Jakobsson and Dhirman's (2013) evaluated the character composition of human-created password. The authors work discovered that linguistic elements are also used to determine the strength of a password. Jakobsson and Dhirman's (2013) discouraged against the use of linguistic elements as they are used by hackers in the creation of effective guessing dictionaries. The work of Mazurek (2013) identified three factors. The first one being that people often reuse their passwords over multiple online accounts which creates a high-valuable target that can be exploited to gain access to these profiles. Secondly, the author stated that the limitation of the human cognition played a huge role in how people create their password. For instance, a person would follow certain patterns to ensure that they can remember their password, such as incorporating names of places or people. Thirdly, Mazurek found that people who manage to create complex passphrases tend to note them down and store them in places that are easily accessible. The findings by Mazurek were supported by a recent survey conducted by Lord (2020). Out of the 10,000 participants surveyed, 18% said they preferred convenience as opposed to security, hence, they created easy passwords. 39% indicated that they wrote down their passphrases on pieces of paper while 10% used computer files, such as dropbox. Lastly, the survey found out that 18.5% of the participants only changed their passwords when notified of a security breach. The literature review also covered how password managers function in securing passwords. Three of the most popularly used password managers, Google, LastPass, and 1Password were discussed in detail.

6.2.2 Objective 2

The second objective was aimed at understanding previous research in password predictive analysis and how machine learning techniques can be used to enhance password creation. Two

existing solutions were discussed in this section, they include John the Ripper and OMEN, a password guessing tool that uses and ordered Markov enumerator. The first one JtR is one of the most popularly known tools used for password guessing through the implementation of brute-force and dictionary attacks. Compared to JtR, OMEN performed faster and efficiently by significantly reducing the amount of time it takes for JtR to produce password guesses. Although both solutions performed predictive analysis on passwords they both had weaknesses. Internet users describe JtR as a complicated system whereby one requires extensive knowledge on how to use the application. On the other hand, OMEN's Markov enumerator was discovered to underperform when compared to either variable or higher-order Markov Model.

6.2.3 Objective 3

The third objective of this study focussed on the design, development and testing of a neural network based password prediction, generation, and storage scheme. Prior to commencement of development, functional and non-functional requirements were gathered and analysed. These requirements were used to design various data flow diagrams including, use case, flow chart, and sequence diagram. The development was done on a personal computer running windows 10 operating system. Model validation and several system tests were conducted to ensure all functional and non-functional requirements had been met before deployment.

6.2.4 Objective 4

The last objective was aimed at validating that predictive analysis on password can be used to improve character composition of human-created passwords. Model validation was conducted using cross-validation technique whereby the model was tested using unseen data. A model accuracy ad loss graph were generated.

6.3 Advantages of the Scheme

6.3.1 Multiplatform

The online password prediction, generation, and storage scheme implemented in this study was developed using python programming language, which is supported by many operating systems. This study focused only on how young learners compose their passwords, the various authentication schemes, and their shortcomings.

6.3.2 Open Source

The neural network based online password prediction, generation, and storage is an open source application that can be downloaded via https://github.com/winniebahati/passwordpredictor.git. Its

distribution was done under the MIT License whose conditions include preservation of copyright and license notices. However, licensed works, modifications, and larger works can be distributed under different terms and without source code.

6.4 Summary

The objectives of the study sought to understand the various factors that predispose passwords to attacks. Consequently, the study pursued an analysis of how humans, construct their passwords. It was established that when individuals are required to create their own secret words, they follow certain linguistic elements that are well known to attackers. Further, it was noted that the constant use of linguistic elements was because users required easy passwords that can be remembered. So as to put all these in perspective, this study explored the use of machine learning techniques, more specifically artificial neural networks as a context in which password prediction and generation can be achieved. The generated complex passwords relied on an embedded password manager to securely store them. The extensive research done in the course of this study, helped inform the need of the proposed scheme. Encouragingly enough, the resulting scheme received positive reviews from the need of having it, to its functionality and usability.

Chapter 7: Conclusion, Recommendation, and Future Work

7.1 Conclusion

The main objective of this study was to develop an online scheme that sought to predict human-created passwords letter-by-letter to completion, in real time, then suggest a stronger computer-generated passphrase within set parameters, such as preferred length, symbols, numbers and lower or uppercase letters. Due to the complexity of the generated password, a password manager will be embedded into the scheme for the purpose of securely storing the new secret word.

A deep learning neural network was modelled to do this. The dataset used was collected from a password dictionary containing two hundred and four thousand passwords. These passwords had a combination of both lower and uppercase letters, special characters, such as @, $, %, &, *, !, and numbers 0 to 9. Similarly, the suggested password was a random computer-generated password that might have a combination of both or one upper and lowercase letters, special characters and numbers but in no particular order.

The model was trained with 80% of the data to predict a user's password, while the remaining 20% used as testing data. The process of cross-validation was used to ensure that the model was able to make correct predictions. A model loss and accuracy graphs were generated during validation. The model's accuracy score stood at 90.3% with a loss score of 0.3. These results validated that predictive analysis on password can be used to improve character composition of human-created passwords.

7.2 Recommendation

To improve on the schemes findings, this study recommends the following:

i. The need for public awareness on the importance of creating passwords that do not rely on known character sets. This scheme can be used to inform Internet users on how predictive analysis of weak passwords can be conducted while improving on their password creation skills.

ii. The open source community is encouraged to test, evaluate, critique and give recommendations on how the scheme can be improved further.

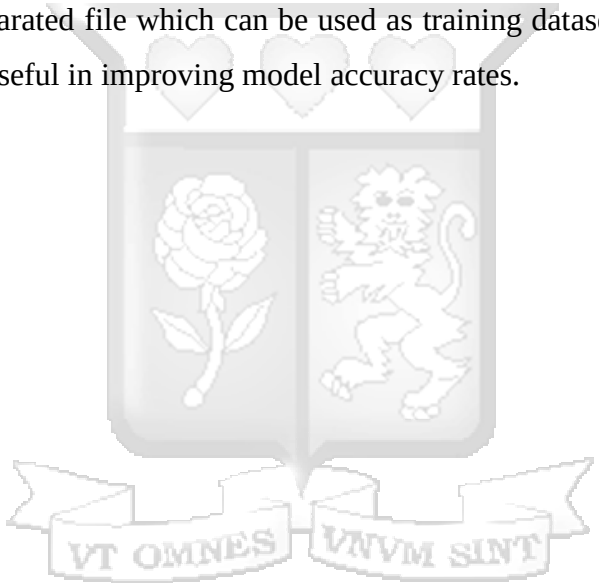
iii. Inclusion of multi-factor authentication processes and password less approach, such as the use of biometrics for instance, fingerprint and image recognition.

7.3 Future Work

Password security is a key area of discussion especially with regards to protecting the online identity of Internet users. Therefore, there is need to maintain an open discussion with all stakeholders involved. As such, some of the future work that could be implemented could be based on the provision of a more interactive platform which can be achieved in several ways, some of which are highlighted below:

i. Implementing a pattern matching algorithm that can be used to offer different multi-factor authentication techniques, such as facial recognition. In this case, online platforms can begin their transformation into password less applications.

ii. After completion of the predictive analysis, the resulting password values can be converted into a comma-separated file which can be used as training dataset in future models. This inclusion can be useful in improving model accuracy rates.



References

Ali, I, A. (2012). Modeling and Simulation of MEMS Components: Challenges and Possible Solutions. Doi: 10.5772/32122.

Anaconda. (2020). Retrieved from: https://www.anaconda.com/ on 10th March, 2021.

Antala, K. (2020, February 13). Why VueJs is so popular for Front-End Development? Retrieved from: https://www.cmarix.com/blog/why-vuejs-is-so-popular-for-front-end-development/ on 6th September, 2021.

Aggarwal, T. (2020, February 26). Fundamentals: Artificial Neural Networks are to deep learning; what atoms are to matter. Retrieved from: https://analysts.com/blogs/fundamentals-artificial-neural-networks-are-to-deep-learning-what-atoms-are-to-matter on 12th April, 2021.

Bansal, S. (2018, March 26). Language Modelling and Text Generation using LSTMs — Deep Learning for NLP. Retrieved from https://medium.com/@shivambansal36/language-modelling-text-generation-using-lstms-deep-learning-for-nlp-ed36b224b275 on 23rd March, 2021.

Barnes, E. (2021). Laravel Tutorial. Retrieved from: https://laravel-news.com/your-first-laravel-application on 6th September, 2021.

Berzerk0. (2018, February 19). GitHub. Retrieved from GitHub: https://github.com/berzerk0/Probable-Wordlists/blob/master/Real-Passwords/WPA-Length/Top204Thousand-WPA-probable-v2.txt on 3th March, 2021.

Baier, L., Jöhren, F., & Seebacher, S. (2019). Challenges in the deployment and operation of machine learning in practice. Retrieved from: https://aisel.aisnet.org/ecis2019_rp/163 on 5th September, 2021.

Bootstrap. (2021), Build fast, responsive sites with Bootstrap. Retrieved from: https://getbootstrap.com/ on 6th September, 2021.

Castelluccia, C., Dürmuth, M., & Perito, D. (2012, February). Adaptive password-strength meters from markov models. In *NDSS*.

Ciampa, M. (2013). A comparison of password feedback mechanisms and their impact on password entropy. *Information Management & Computer Security*. Vol 21, (5). Doi: 10.1108/IMCS-12-2012-0072

Cimpanu, C. (2019, August 27). Microsoft: Using multi-factor authentication blocks 99.9% of account hacks. Retrieved from: https://www.zdnet.com/article/microsoft-using-multi-factor-authentication-blocks-99-9-of-account-hacks/ on 5th February, 2021.

CISOMAG. (June, 2020). 8Belts Exposes PII of e-learners in a Data Breach. Retrieved from: https://cisomag.eccouncil.org/8belts-exposes-e-learners-data/ on 5th February, 2021.

Chestwick, W. (2013, February 1). Rethinking Passwords. *Communications of the ACM*. Doi: 10.1145/2408776.2408790.

Ching, Y., Hsu, Y., Baldwin, S. (2018, April 30). Developing Computational Thinking with Educational Technologies for Young Learners. Doi: 10.1007/s11528-018-0292-7.

Colin, N. (2018). The Benefits of Using Bootstrap for Front-End Development. Retrieved from: https://digitalmad.co.uk/the-benefits-of-bootstrap-for-front-end-development/ on 6th September, 2021.

Dell'Amico, M., Michiardi, P., and Roudier, Y. (2010). Password strength: *An empirical analysis. In INFOCOM'10*. Proceedings of the 29th Conference on Information Communications, IEEE, 983–991. Doi: 10.1109/INFCOM.2010.5461951.

DeepAI. (2019, May 17). Vanishing Gradient Problem. Retrieved from DeepAI: https://deepai.org/machine-learning-glossary-and-terms/vanishing-gradient-problem on 5th February, 2021.

Django. (2021). Documentation. Retrieved from: https://www.djangoproject.com on 13th March, 2021.

Farash, M. S., Islam, S. K. H., & Obaidat, M. S. (December 10, 2015). A provably secure and efficient two-party password-based explicit authenticated key exchange protocol resistance to password guessing attacks. Concurrency and Computation: Practice and Experience, 27, 17, 4897-4913. Doi: 10.1002/cpe.3477.

Harandi, S. R. (2015, May). Effects of e-learning on Students' Motivation. *Procedia-Social and Behavioral Sciences, 181*, 423-430. Doi: 10.1016/j.sbspro.2015.04.905.

Imperva. (2021). Two Factor Authentication (2FA). Retrieved from: https://www.imperva.com/learn/application-security/2fa-two-factor-authentication/ on 5th April 2021.

Jakobsson, M., & Dhiman, M. (2013). The benefits of understanding passwords. In *Mobile Authentication* (pp. 5-24). Springer, New York, NY. Doi: 10.1007/978-1-4614-4878-5_2.

Jancis, M. (2021, March 30). How do password managers work? Retrieved from: https://cybernews.com/best-password-managers/how-do-password-managers-work/ on 12th April, 2021.

Johnson, R, D., & Brown, K, G. (2017, August). E-Learning. The Wiley Blackwell Handbook of the Psychology of the Nternet at Work, 369-400. Doi: 10.1002/9781119256151.ch17.

Joby, A. (2021, July 21). What is Cross-Validation? Comparing Machine Learning Models. Retrieved from: https://learn.g2.com/cross-validation on 5th September, 2021.

Keane, J. (2017, March 22). Latest bugs in LastPass allowed attackers to steal passwords. Retrieved from: https://www.digitaltrends.com/computing/lastpass-bugs-password-ormandy/ on 30th April, 2021.

Kaspersky. (2020, September 4). Digital Education: *The cyberrisks of the online classroom*. Retrieved from: https://securelist.com/digital-education-the-cyberrisks-of-the-online-classroom/98380/ on 15th April, 2021.

Komanduri, s., Keley, p., Mazurek, M., Shay, p., Vidas, T., Christin, N., Bauer, L., Cranor, L., Lopez, j. (2012). Guess Again (and Again and Again): Measuring Password Strength by Simulating Password Cracking Algorithms. Doi: 10.1109/SP.2012.38.

Kostadinov, S. (2019, August 8). Understanding Backpropagation Algorithm. Retrieved from Towards data science: https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd on 5th February, 2021.

Keras. (2021). Retrieved from: https://keras.io/getting_started/ on 10th March, 2021.

Kenton, W. (2020, September 28). Two-Factor Authentication. Retrieved from: https://www.investopedia.com/terms/t/twofactor-authentication-2fa.asp on 5th February, 2021.

Krol, K., Philippou, E., De Cristofaro, E., & Sasse, M. A. (2015, January 19). "They brought in the horrible key ring thing!" Analysing the Usability of Two-Factor Authentication in UK Online Banking. *arXiv preprint arXiv:1501.04434*.

Landman, M. (2013, June 28). What is single-factor authentication? Retrieved from: https://www.rippleit.com/blog/what-is-single-factor-authentication on 5th February, 2021.

Laravel. (2021). Laravel Documentation. Retrieved from: https://laravel.com/docs/8.x/readme on 6th September, 2021.

Lord, N. (2020). Uncovering Password Habits: *Are User's Password Security Habits Improving? (Infographic*). Retrieved from: https://digitalguardian.com/blog/uncovering-password-habits-are-users-password-security-habits-improving-infographic on 10th January 2021.

Ma, J., Yang, W., Luo, M., & Li, N. (2014, May). A study of probabilistic password models. In *2014 IEEE Symposium on Security and Privacy* (pp. 689-704). IEEE. Doi: 10.1109/SP.2014.50.

Marhon, S. A., Cameron, C. J., & Kremer, S. C. (2013). Recurrent Neural Networks, Handbook on Neural Information Processing. Intelligent Systems Reference Library, vol 49, pp. 29-65. Doi: 10.1007/978-3-642-36657-4_2.

Martindale, J. (2021). LastPass vs. 1Password. Retrieved from: https://www.digitaltrends.com/computing/lastpass-vs-1password-comparison/ on 1st May, 2021 .

Mazurek, M. L., Komanduri, S., Vidas, T., Bauer, L., Christin, N., Cranor, L. F., Shay, R., (2013, November). Measuring password guessability for an entire university. Proceedings of the Acm Conference on Computer and Communications Security, 173-186. Doi: 10.1145/2508859.2516726.

Miessler, D. (2019, August 18). SecList. Retrieved from: github.com/danielmiessler/SecLists on 3rd March, 2021.

Mills, M. (2021, August 15). How to Crack Password or Keys Very Fast Using John the Ripper. Retrieved from: https://itigic.com/crack-passwords-or-keys-very-fast-using-john-the-ripper/ on 5th September, 2021.

Netmux. (2020). Weakpass. Retrieved from: https://weakpass.com/ on 3rd March, 2021.

Nilesh, A, L., Salendra, P., Mohammed, F. (2016, November 11). A Review of Authentication Methods. *International Journal of scientific & technology research. Vol 5, (11).*

NumPy. (2020). NumPy. Retrieved from: https://numpy.org/ on 20th April, 2021.

Olah, C. (2015, August 27). Understanding LSTM Networks. Retrieved from: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ on 18th February, 2021.

Ometov, A., Bezzateev, S., Makitalo, N., & Andreev, S. (2018, January). Multi-Factor Authentication: A Survey. Doi: doi:10.3390/cryptography2010001.

Pallets. (2010). Retrieved from: https://flask.palletsprojects.com/en/1.1.x/ on 10th March, 2021.

Pandas. (2020, March 18). Pandas. Retrieved from: https://pandas.pydata.org/ on 3rd March, 2021

Pandey, P. (2019, November 25). Data Pre-processing: Concepts. Retrieved from: https://towardsdatascience.com/data-preprocessing-concepts-fa946d11c825 on 3rd March, 2021.

Password Manager. (2021). What is a Password Manager? Retrieved from: https://www.passwordmanager.com/what-is-a-password-manager/ on 28th April, 2021.

Pedamkar, P. (2020). RAD Model. Retrieved from: https://www.educba.com/rad-model/ on 5th September, 2021.

Pedamkar, P. (2020). Loss Function in Machine Learning Retrieved from: https://www.educba.com/loss-functions-in-machine-learning/ on 6th September, 2021.

Perakath, B., Mukul, P., Richard, M. (2006). Using Ontologies for Simulation Modelling. *Proceedings of the 2006 winter simulation conference.* Doi: 10.1109/WSC.2006.323206.

Plötz, T., & Fink, G. A. (2011). Markov Model Concepts: The Essence. In *Markov Models for Handwriting Recognition* (pp. 19-26). Springer, London.

Rosencrance, L. (2018, May). Authentication. Retrieved from: https://searchsecurity.techtarget.com/definition/authentication on 5th February, 2021.

Salimian, R, H. (2017, November 24). Recommendation Systems in Machine Learning. Retrieved from: https://www.zeolearn.com/magazine/recommendation-systems-in-machine-learning on 05th February, 2021.

Sargent, R. (2011). Verification and Validation of Simulation Models. *Proceedings of the 2011 Winter Simulation Conference*.

Singh, A. (2019, December 6). What is Rapid Application Development (RAD)? Retrieved from: https://blog.capterra.com/what-is-rapid-application-development/ on 5th February 2021.

Spyder. (2020). Retrieved from: https://www.spyder-ide.org/ on 10th March, 2021

TensorFlow. (2020, March). TensorFlow. Retrieved from https://www.tensorflow.org/ on 3rd March, 2021

Terrazzoni, J. (2019, June 7). Implementing the SHA256 and MD5 hash functions in C. Retrieved from: https://medium.com/a-42-journey/implementing-the-sha256-and-md5-hash-functions-in-c-78c17e657794 on 8th April, 2021.

Vue.Js. (2021). VueJs Introduction. Retrieved from: https://vuejs.org/v2/guide/ on 6th September, 2021.

Wilcox, M. (2021, May 13). What is one of the disadvantages of using John the Ripper? Retrieved from: https://colors-newyork.com/what-is-one-of-the-disadvantages-of-using-john-the-ripper/ on 5th September, 2021.

Williams, C. (2021, March 20) Data leaks in online education: Almost 1 million records exposed. Retrieved from: https://www.wizcase.com/blog/educational-breaches-research/ on 20th April, 2021

Appendices A: Survey Questionnaire

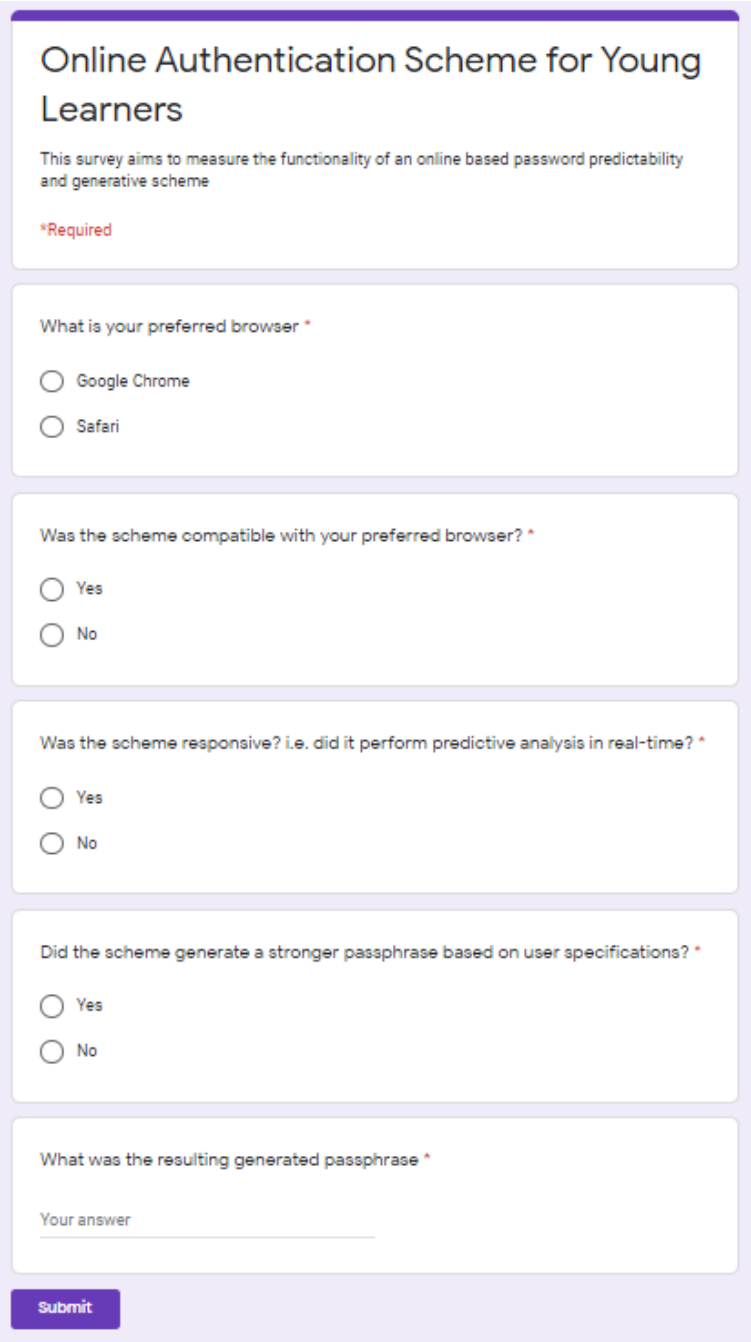


Figure A.1 Survey Questionnaire

Appendices B: Participant Responses

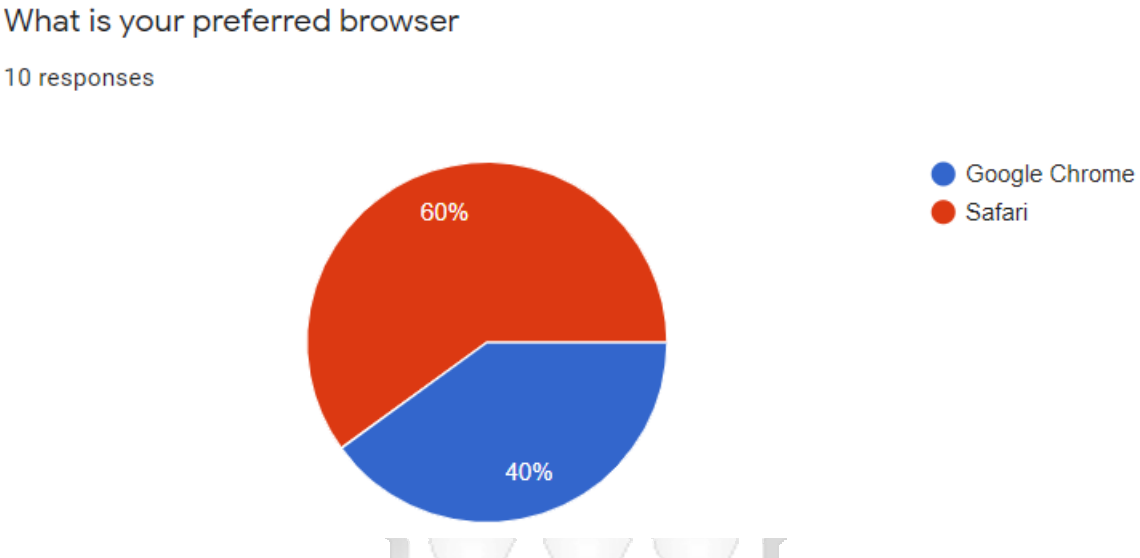


Figure B.1 Question 1

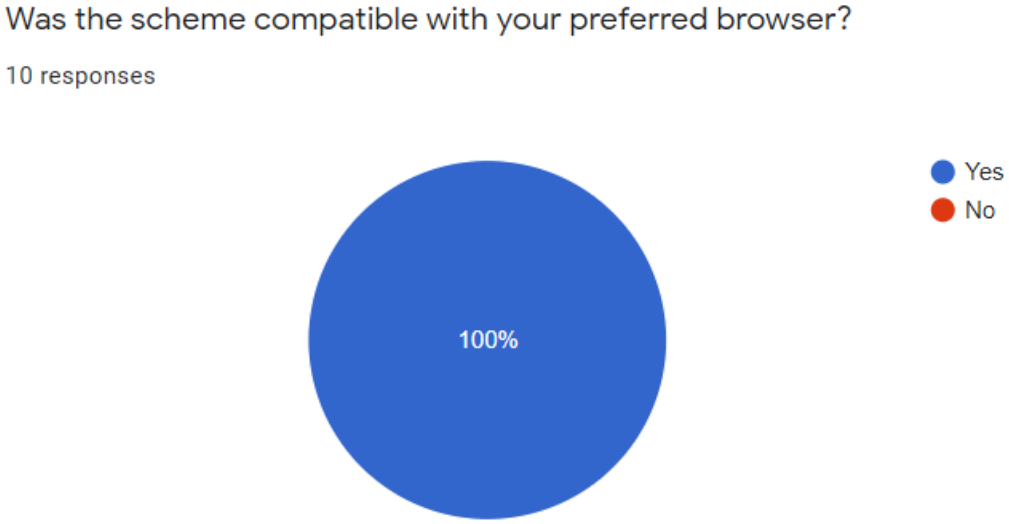


Figure B.2: Question 2

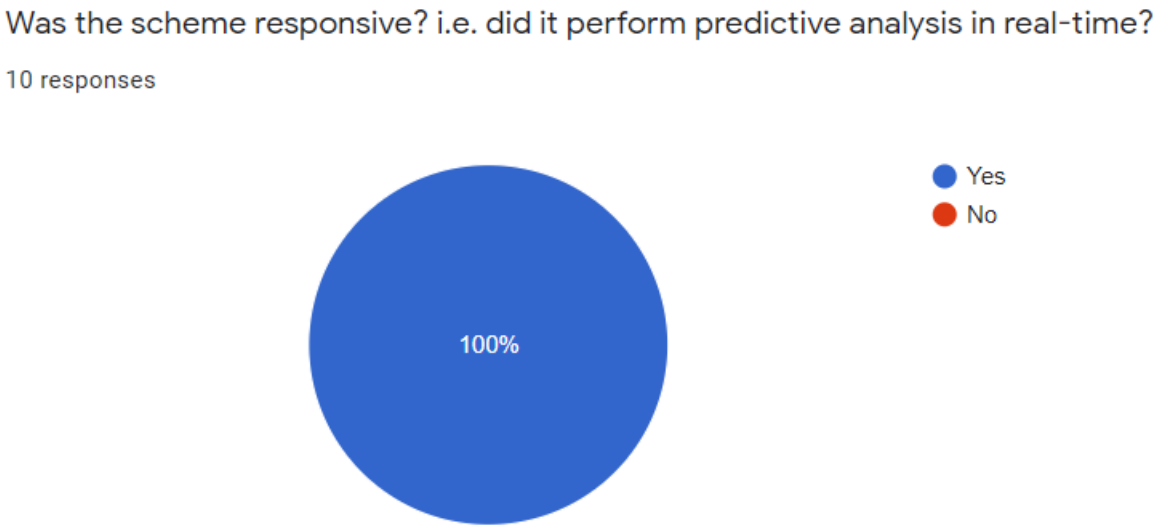


Figure B.3: Question 3

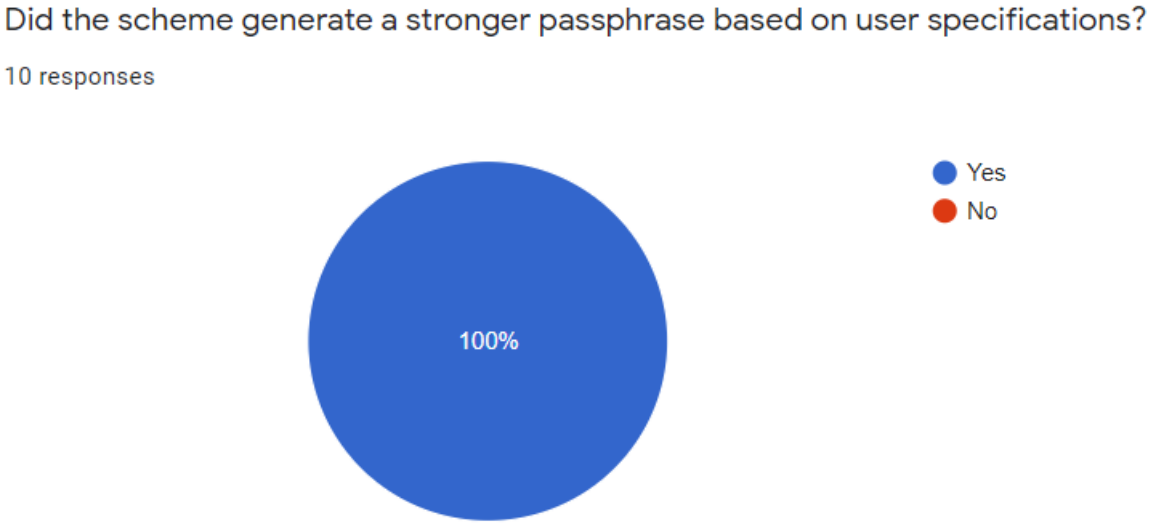


Figure B.4: Question 4

Appendices C: Plagiarism Report and Ethical Submission

Curiginal

Document Information

Analyzed document	A Neural Netwok based Authentication Scheme for Young Learners.pdf (D109859467)
Submitted	6/28/2021 8:01:00 PM
Submitted by	
Submitter email	winniebahati@gmail.com
Similarity	2%
Analysis address	library.strath@analysis.urkund.com

Figure C.1: Plagiarism Report

RHInnO Ethics - SU-IERC1089/21 - 1 of 1

Completion of Online Research Ethics Review Submission

You have successfully submitted your application for ethics review "A neural Network Authentication Scheme for Young Learners"
Certificate awarded to: Ms Bahati, Winnie
Reference number: SU-IERC1089/21
Date and Time: 2021-06-16 10:14:18

Figure C.2: Ethical Submission