

Identifying Anomalies in MFA with AI

Dissertation Report



April 5, 2022

Qumer Arshad

1954609

Table of Contents

[Table of Figures 1](#_Toc103441058)

[Abstract 2](#_Toc103441059)

[Introduction 2](#_Toc103441060)

[Literature Review 3](#_Toc103441061)

[Risk-based Authentication 3](#_Toc103441062)

[Biometric Approach to MFA 6](#_Toc103441063)

[Implementation 6](#_Toc103441064)

[The 2FA Playground 6](#_Toc103441065)

[Machine Learning Model 12](#_Toc103441066)

[Data Preprocessing 12](#_Toc103441067)

[Results, Analysis and Evaluation 15](#_Toc103441068)

[Conclusion 16](#_Toc103441069)

[Recommendations 16](#_Toc103441070)

[References 17](#_Toc103441071)

# Table of Figures

[Figure 1 - Login Page. 10](#_Toc103383971)

[Figure 2 - Get User Request IP. 11](#_Toc103383972)

[Figure 3 - Get User Request Country of Origin. 11](#_Toc103383973)

[Figure 4 - Send OTP to the User. 12](#_Toc103383974)

[Figure 5 - User enters Received OTP. 12](#_Toc103383975)

[Figure 6 - Correct OTP and Normal User Request. 12](#_Toc103383976)

[Figure 7 - Either Wrong OTP or Anomalistic User Request. 13](#_Toc103383977)

[Figure 8 - .txt Log File. 13](#_Toc103383978)

[Figure 9 - Sample Training Data. 13](#_Toc103383979)

[Figure 10 - Total Data Count and Null Value Count. 14](#_Toc103383980)

[Figure 11 - Selecting Not Null Instances and Reinitializing the Dataframe. 14](#_Toc103383981)

[Figure 12 - Null Count, Dataframe Length and Unique Count. 14](#_Toc103383982)

[Figure 13 - Normal Count vs Anomaly Count 15](#_Toc103383983)

[Figure 14 - Label Encoding the Dataset. 15](#_Toc103383984)

[Figure 15 - Splitting the Dataset into Testing and Training Sets. 16](#_Toc103383985)

[Figure 16 - Model Fitting, Performance Metrics and Model Pickling. 16](#_Toc103383986)

[Figure 17 - Project Plan 20](#_Toc103383987)

[Figure 18 - Gantt chart 21](#_Toc103383988)

[Figure 19 - Known Risks and Risk Management 21](#_Toc103383989)

[Figure 20 - Gibbs’ Reflective Cycle (Gibbs 1988) 23](#_Toc103383990)

# Abstract

Two-factor authentication has been the university standard for securing user authentication all around the world. But with the advent of better technology with a rapid pace, 2FA has started to fail in some use-cases and these scenarios are occurring more often. The aim of this research is proposing a novel approach that involves adding a 3rd layer of protection provided by Machine learning to verify the integrity of user requests on a website in between the process of submitting the user credentials and entering the One-time Password that the user receives as a feature of 2FA.

Keywords: **Machine Learning, 2FA, User Request Integrity**

# Introduction

With the advent of technology, the security and safety of our electronic lives are not far-fetched. The only problem is that the development in security technology is a double-edged sword due to the freedom of access people have to technology. Thus, hackers also get their hands on this technology easily and find ways to get past it. Two-factor authentication (2FA) is one of the most innovative security measures to prevent hackers and malice from approaching our social accounts. 2FA requires a third-party app or device to verify the account details during login.

Cell phones assume significant parts in numerous individuals' day-by-day life. Individuals usually use cell phone applications to take photographs, send messages, book rides, or shop on the internet. It is not strange for those applications to ask for private data (like names, sex, or Visa data) from their clients to improve the nature of their administration. The delicate idea of those private data requires application engineers appropriately tie-down admittance to their administration. A mainstream approach to getting such access is by asking for passwords from clients during login measures (Irvan et al., 2021).

In any case, passwords and other information-based validation strategies like PIN (individual ID number) codes convey extraordinary danger as clients will in general utilize similar passwords across different administrations. Accordingly, numerous administrations right now require extra belonging-based authentication techniques before allowing access. An ordinary method of this execution is by sending an interesting code through SMS (short message administration) to clients' telephone numbers. This additional progression is 2-factor validation (2FA) or multifaceted confirmation (MFA). Tragically, ownership-based validation techniques carry possible bothers to clients since e they may need to convey extra gadgets, which can be effectively lost. Numerous clients additionally utilize the same cell phone to enter passwords and get 2FA codes. In this manner, if their cell phone goes missing, assailants can sidestep 2FA checks (Irvan et al., 2021).

To prevent this, a novel solution of creating a third layer in the 2FA that will detect anomalous attempts using machine learning has taken its bearings for implementation. This also claims the title 3FA or the 3-Factor Authentication for ease of discussion in this paper.

In April 2019, Kaspersky researchers revealed an enormous scope of SIM trade misrepresentation activities focusing on clients in both the Portuguese-talking countries of Brazil and Mozambique who had the option to utilize social designing, pay off and straightforward phishing assaults eventually taking cash from casualties. Danger entertainers did these assaults by assuming responsibility for a casualty's telephone number by capturing accounts and catching two figure confirmation strategies in which the subsequent authentication factor is an SMS message or a call put to the portable number.

Two-factor validation, the additional security step that requires individuals to enter a code shipped off their telephone or email, has generally attempted to protect usernames and passwords from phishing assaults. In any case, security specialists have shown a mechanized phishing assault that can slice through that additional layer of safety—likewise called 2FA—conceivably fooling clueless clients into sharing their private qualifications.

Cases like these are popping up all over the country. Various hackers have compromised the integrity of the 2FA security. Researchers are trying to keep up with their security measures but the requisite of something intelligent and robust still exists.

# Literature Review

Authentication holds a mandatory place in the Cybersecurity Domain (Townsend, 2021). With the advent of MFA (Multi-factor Authentication), security has improved rapidly. However, access to new technology is not limited to good people only. Cybercriminals are constantly evolving their methods and strategies including adding Artificial Intelligence to their list of tools (Townsend, 2021). Authentication is an early and basic line of safeguard for business information. Yet, traditional authentication because of passwords stays a flimsy part. That is because clients are famous for awful password practices (Townsend, 2021):

* Reusing passwords
* Utilizing unsurprising ones
* Putting away secret key data on tacky notes or in decoded bookkeeping sheets

Subsequently, an ever-increasing number of organizations are adding MFA (Multi-factor Authentication), in any event, for their client encounters, requiring clients to make an item buy or go through with a bank exchange from their telephone for additional authentication. By adding extra factors past the secret word or, far and away superior, instead of the secret key organizations can assist with defeating secret word shower and social designing assaults and stop hackers utilizing taken certifications from truly entering the record. Extra factors could incorporate addressing a security question, utilizing a one-time secret word, or answering a pop-up message on the telephone (Townsend, 2021).

Hackers are shrewd, however, and even with MFA, individuals' records might break and leak (Townsend, 2021). Gadgets, similar to telephones or USBs, are prone to stealing. OTPs (One-time Passwords) communicated through SMS are easily captured. What is more, biometrics, like fingerprints and, surprisingly, facial acknowledgement, can be hacked or faked. As Artificial, intelligence gets forward movement; it turns out to be considerably more straightforward to counterfeit even biometrics, making counterfeit fingerprints and facial pictures with an adequate number of matching focuses to pass an output. However, MFA improvement is possible by adding a basic snippet of data: context. Context is the data about the client's login, similar to where the client is while endeavouring to sign in or the gadget used. Such context can give basic insights that an assault is occurring (Townsend, 2021).

## Risk-based Authentication

To add context, the Identity and Access Management (IAM) industry has answered with risk-based authentication. Standard MFA catches data about what the client knows, similar to a password, what the client has, similar to their telephone, and even who the client is utilizing biometrics like fingerprints. Risk-based validation takes into account extra factors that help decide whether the client truly is who they say they are (Townsend, 2021). This is finished by contrasting their past login conduct with the present authentication endeavour, giving setting data that is absent in standard MFA. For instance, assuming a client ordinarily signs in on a specific PC from the fundamental office area during the week but abruptly attempts to sign in from a telephone at Starbucks, it could be an indication of a taken PC or compromised account.

On the other hand, assuming the client commonly signs in from home through one IP address and out of nowhere is signing in from another IP address (Townsend, 2021). Maybe one on a rundown of dubious IPs - you would need to challenge the login endeavour and request an extra validating component like a one-time secret word or face examination from a symbolic gadget. Evaluating authentication information like this continuously requires escalated and complex handling. That is where Artificial Intelligence comes in that place.

To execute risk-based authentication, cybersecurity organizations use AI-upheld innovations. The AI surveys and weighs individual variables about the login endeavour to think of a gamble score for the situation. For instance, a client associating with specific IP addresses or endeavouring to sign in during the centre of the night could show a danger (Townsend, 2021).

Artificial intelligence can likewise involve brain networks as a component of AI frameworks. These brain networks emulate the human mind and "learn" by being taken care of by datasets that incorporate the right outcome (Townsend, 2021). For instance, information about signing in utilizing various IPs and the outcomes demonstrating which of those logins were cyberattacks. It resembles a child offered a variable based math issue and the response, who should sort out what the recipe is to take care of this kind of variable based math issue. The AI grows endlessly better calculations to figure out which variables demonstrate an assault by attempting various methods to take care of the issue and looking at its response against the response in the dataset (Townsend, 2021). In the end, it tracks down a bunch of calculations that help anticipate dangers more often than not.

Risk engines screen various elements in a client's logins over the long run and assemble a profile for every client to comprehend login designs. At the point when a client differs from that profile on a given validation endeavour, the AI framework surveys the variable factors and decides a gamble score for the current login endeavour. A portion of the variables normally represented includes (Townsend, 2021):

* Network reputation
* Client's geographic area
* The device fingerprint (like the producer, model, or program)
* Login Time

While the vital advantage of AI-fueled risk-based confirmation is security, it can likewise smooth out the validation cycle. In standard MFA, clients are provoked for extra factors at each login endeavour. Enter your username and secret key, and then, at that point, answer a security question. Then again, enter your username and secret key, and then answer a message pop-up on your telephone. With AI-fueled confirmation, clients at generally safe probably will not be requested any extra factors, making login quicker (Townsend, 2021).

Risk-based authentication will proceed to improve and get more brilliant. At last, risk-based authentication will probably move from directed realizing, where the dataset incorporates the results, to solo realizing where the AI observes new examples that people might not have found and makes forecasts of expected variables to survey (Townsend, 2021). Having the option to cross-reference different AI calculations and use design acknowledgement and time-series, based prescient calculations will work on the precision and extent of AI-based validation contributions going ahead, for internet application logins, yet in addition for different parts of online protection like organization interruption and botnet recognition.

Simultaneously, designers will be searching for ways of giving IT divisions more command over the AI framework, for example, the capacity to see precisely why the AI settled on a given choice, change the number of perceived variables, and tailor the framework to their association's special climate. Albeit not stringently AI, different cross-industry drives are in progress to empower better information sharing so the data one association has on a potential danger can be made accessible to different associations progressively, further developing MFA (Townsend, 2021).

You can likewise hope to see AI-fueled validation frameworks extend to envelop consistent confirmation. Rather than constant danger evaluation exactly at login, AI frameworks will recognize and answer dangers all through a client meeting (Townsend, 2021). Assuming the client unexpectedly moves to another area and gadget, or endeavours to get to monetary data that is not pertinent to their work, the prompt to confirm their personality comes up.

In addition, surprisingly, further, access to the board will probably move from the application level to the information level. Specialists are now looking at appending metadata to individual bits of information, to demonstrate who ought to have what sort of admittance to that discrete snippet of data. For instance, the field in a data set containing worker compensations would include metadata demonstrating that main clients inside the organization who hold specific jobs can see that data (Townsend, 2021). At the point when that compensation data is exposed, the limitations on access arise. Artificial intelligence-controlled validation would then implement these information level access limitations in any place the information is utilized.

As the character and access to the executives’ necessities advance, so will AI as an instrument in IAM. Since, the truth of the matter is, that AI is important to deal with the intricacy of investigation at the scale and speed that will be required in the changing danger scene and the developing character and access to the board climate (Townsend, 2021).

MFA utilizes any mix of at least two elements to confirm personality and keep crucial resources secure from fake access. At this point, we've all pre-owned two-factor validation (2FA) online to approve a login or exchange by joining a secret word with an SMS code shipped off our cell phone. In the event of the compromise of a component, the framework is yet secure (Kightlinger, 2019).

Three primary elements take the place in play to affirm character (Kightlinger, 2019):

* Something you have – a physical item, for example, an ATM card, key fob or USB stick.
* Something you know - "confidential" like a password or PIN.
* Something you are - a biometric element, for example; fingerprints or voice, iris scans and other physical attributes.

The guidelines for how to consolidate these elements and use them to verify character rely upon the element carrying out them. In specific businesses, Current expectations from MFA require it to meet consistency commands. For instance, the Payment Card Industry Data Security Standard (PCI DSS) requires MFA for character and access to the executives in unambiguous conditions - for example, remote admittance to a cardholder information climate that begins from outside the organization or administrator admittance to the information climate from inside the confided in the network (Kightlinger, 2019).

An ever-increasing number of associations are thinking about consistently on MFA for each application and IT framework, yet all the same, that is quite often excessively lumbering. Assuming representatives need to hang tight for an SMS code to arrive at their telephone each time they need to get to an application, client purchases decrease with time (Kightlinger, 2019). A more compelling way to deal with getting the venture includes strategies to lessen the weight of activity on the client however much as could be expected and focusing on the applications that require 2FA in light of awareness and chance of giving and taking.

Indeed, even where MFA is justified, nonintrusive gamble based or context-oriented confirmation can make it less baffling for clients. Nonintrusive validation factors incorporate gadget fingerprinting, geolocation, IP, gadget reputation, and portable organization administrator information (Kightlinger, 2019). Some danger insight stages, for example, the IBM X-Force Exchange, as of now give this data to outsider applications and arrangements.

These components add the setting to the client and gadget for an exchange and assist with evaluating the gamble level of every activity. If the gamble is too extraordinary, extra confirmation is required. For example, assuming a client in New York signs in to the corporate organization utilizing her work area, you may not need MFA; yet if a client in Hong Kong attempts to get to an application through an obscure organization utilizing an unnoticed gadget, you certainly need to add validation measures (Kightlinger, 2019).

Stages incorporating fraud detection technologies and unified endpoint management (UEM) tools assist with diminishing the requirement for client-driven MFA and give accommodating settings about the client's gamble level to decide the requirement for extra layers of verification (Kightlinger, 2019). Such stages engage associations to oversee and get every one of the numerous ways representatives to an interface when they are portable, for example, cell phones, PCs, wearables and even web of things (IoT) gadgets. An open stage likewise makes coordination with existing applications and foundations direct.

MFA might be fine for workers, who can be expected to utilize anything confirmation instrument their association picks. Organizations have customarily weighed security versus accommodation, continuously underscoring the last option out of dread that clients would dismiss additional means to safeguard individual information (Kightlinger, 2019).

In any case, this customary way of thinking may presently not be valid as we are seeing an expanded degree of acknowledgement and experience with multifaceted confirmation from everyone. Truth be told, as the "IBM Future of Identity Study 2018" showed, shoppers have become more natural and tolerant of MFA, particularly concerning cash related applications and virtual entertainment. Contingent on the age bunch, the sort of MFA favoured differs, with the more youthful age substantially more OK with cell phone innovation and biometric techniques or tokens as opposed to passwords (Kightlinger, 2019).

Organizations could observe that the ideal arrangement is to give clients a decision among different verification choices, whether that is one-time passwords or fingerprint readers (Kightlinger, 2019). Risk-based approaches similar to those for employees become usable in access scenarios for consumers. As the likely mischief from unusual movement rises, so can the number of confirmation factors required.

Techniques for MFA are constantly changing as weaknesses emerge; innovation advances and the predominant players progressively come from the millennial and Gen Z populaces (Kightlinger, 2019). New MFA approaches should supplant bulky logins with captivating, super-advanced conceivable outcomes. Brilliant organizations will remain adaptable and versatile by using a cloud stage that updates with the most recent techniques.

Moreover, dealing with picking an MFA strategy/technique is now like a piece of information-driven analysis - security pioneers ought to shift focus over to stages that permit them to screen the achievement paces of their verification techniques (Kightlinger, 2019). The strategies and arrangements you carry out today will not and ought not to be extremely durable. Gathering information persistently will assist you with formulating multifaceted validation techniques that give the ideal security, accommodation and complexity for your workers, clients and organization.

## Biometric Approach to MFA

To upgrade the security of corporate frameworks, for example, cell phones, security elements are conveying MFA to add an extra layer of insurance. MFA depends on three variables:

1. Something You Know (e.g., a secret key)
2. Something You Have (e.g., a symbolic gadget or Short Message Service (SMS))
3. Something You Are (e.g., fingerprints or voiceprints).

Indeed, even with the organization of MFA, it is yet conceivable to sidestep the MFA validation process. Consider the situation wherein there is a cell phone in possession (Something You Have). If security does not claim the protection of the cell phone and does not have a screen lock, the hoodlum can reset the secret phrase on financial applications, or even get client data from the gadget (Something You Know). A Phishing assault is one more method for taking a client's qualifications, where the casualty opens an email or message and clickbait fools the user into tapping on a noxious connection, which can prompt getting the Something You Have and Know. Indeed, even dependence on (Something You Are) is risky. For instance, one can utilize a general fingerprint (genuine or engineered fingerprints that can serendipitously coordinate with an enormous number of fingerprints [5]) which can break 65% of genuine fingerprints [6], or one can utilize the casualty's biometric examining, for example, a fingerprint, voice recording or photograph, which can be acquired from the gadget. To stay away from the recently referenced Phishing assaults, an individual's uniqueness has been utilized as verification apparatus. A client's mark is an instrument utilized for client verification and has been in need for a long time. Even more as of late, marks have arisen as a biometric acknowledgement apparatus. The client needs to give their unmistakable, which drives them to put away information in the framework. Nonetheless, if the put away biometric information were to be compromised, the information would be significant to the hacker's hands (e.g., fashioning marked documents can be utilized). This pushes us to ponder a more secure method for joining the client's interesting credits to concede admittance to delicate information on their cell phones.

# Implementation

The project has two basic modules. The modules are as follows:

## The 2FA Playground

The 2FA playground will be, as the name suggests, a playground for testing and experimenting in this project. The playground will be enabled with two main features i.e., the 2FA auth. and request tracking. The language used for this research project will be Python, and for the development of this 2FA playground which is basically a web application, Flask, a python web framework will be employed. The reason behind using Python for the complete project is because it is much easier to implement Machine Learning models on Python with the currently available vast collection of easy-to-use libraries and tools for Machine Learning. Since the Machine Learning Model is implemented in Python, to integrate the model with a web interface will be a lot more forgiving if the web interface itself was built out of Python.

The 2FA Playground provides a platform where the user can edit and make changes in the code or change the Machine Learning model or the dataset accordingly while providing a self-built dummy 2FA verification system independent of any third-party interreferences to keep the results and performance as unbiased and efficient as possible. The website uses simple HTML pages only containing a form that take in the user’s phone number and send him a verification code to verify it on the other form. The application uses page routing to provide other useful information for debugging as well. The code is as follows:

# OTP Generator Helper Function

*def* generateOTP():

    return random.randrange(10000, 99999)

The above function is a helper function. This function is used in the code to generate a random OTP every time a user enters his phone number.

# OTP API Helper Function

*def* getOTPApi(*phone*):

    account\_sid = os.getenv("TWILIO\_SID")

    auth\_token = os.getenv("TWILIO\_AUTH\_TOKEN")

    otp = generateOTP()

    session['response'] = str(otp)

    client = Client(account\_sid, auth\_token)

    message = client.messages.create(*to*=*phone*,

*from\_*="+19793664234",

*body*="Your OTP is: " + str(otp))

    if message.sid:

        return True

    else:

        False

The above helper function is the one which enables sending the SMS text with the OTP to the user. This function has a few functions namely:

1. It initializes and connects with Twilio, a third-party service that provides a dummy phone number to send mass SMS texts. Through this service, the application will be able to send the OTP SMS text to the user.
2. It the generates a random OTP code using the generateOTP() helper function.
3. Then it stores the randomly generated OTP in the application session for the user. This will be used in the future to verify whether the user has entered the correct OTP or not.
4. Then it sends the message to the user.
5. It then returns ‘True’ if the message has been sent or ‘False’ if not.

# Get Client IP Address

*def* client\_ip():

    ip = request.headers.getlist("X-Forwarded-For")[0]

    return ip

The above helper function is for getting the user IP address from the user request. Since the application is to be hosted on a free Heroku server, a request header has been initialized that will return the user IP from the request.

# Get Client Country

*def* client\_country():

    ip = client\_ip()

    response = requests.get("http://ip-api.com/json/{}".format(ip))

    js = response.json()

    country = js['countryCode']

    if country:

        return country

    else:

        return "User IP Not Accessible!"

The above helper function is used to get the country of origin of the user request from which the IP has been taken. The function takes in the user IP returned from the client\_ip() helper function. The function then sends this IP address to a free online API that returns the country of origin of the IP address by analysing it. It then returns the country code of the user request.

# Import Model and Predict

*def* model():

    # Loading ML Model

    model = joblib.load('./model/model.joblib')

    # Getting Client IP

    ip = client\_ip()

    # Getting Client Country

    country = client\_country()

    # Creating DataFrame for ML Model

    x = {'IP': [ip], 'Country': [country]}

    user\_info = pd.DataFrame(x)

    categ = ['IP', 'Country']

    # Encoding DataFrame

    le = LabelEncoder()

    user\_info[categ] = user\_info[categ].apply(le.fit\_transform)

    # Fitting Model

    prediction = model.predict(user\_info)

    # Writing Prediction Result in a .txt File

    prediction\_file = open("prediction\_file.txt", "a")

    prediction\_file.write("\n")

    prediction\_file.write(str(prediction))

    # Returning Prediction Result

    return prediction[0]

Perhaps the most important helper function of the 2FA web application is the above function. The above function is the one that does all the heavy lifting, i.e., provides an intelligent and secure ML Model layer to verify the integrity of the user request before 2FA takes place. The function performs quite a few functions:

1. It loads in the pickled ML Model into the web application.
2. It gets the User IP and Country of Origin from the above-mentioned helper functions.
3. Since the ML Model input needs to be calibrated for the model to be able to easily digest, the data from the helper functions i.e., the User IP and the Country of Origin are compiled into a DataFrame.
4. Since the ML model cannot understand “string” values, the created dataframe is then encoded using the Label Encoder from scikit-learn library.
5. Now that the data is ready, the model.predict(user\_info) is called passing the finalized dataframe as an argument.
6. The prediction results are then logged into a .txt file that is created locally on the server.
7. The prediction result is also returned for that instance. The returned result can be used to then verify whether the user was a ‘normal’ user or did it belong to the ‘anomaly’ class.

After the implementation of the helper functions, it is time to implement the page routing to easily structure the application and make it scalable-ready for the future. Since the web application is just a playground and a proof-of-concept, there is not styling or CSS elements, just HTML and Python. This website is also hosted on the free Heroku server. To access the website, follow the link: [2FA Playground - Login (cyber-ml.herokuapp.com)](https://cyber-ml.herokuapp.com/).

@app.route("/")

*def* home():

    return render\_template("login.html")

The above route is the homepage of the web application which also houses the login form where the user enters his phone number to receive an OTP.

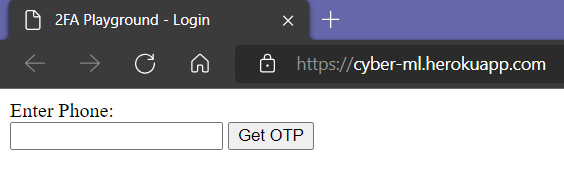


Figure 1 - Login Page.

@app.route("/ip")

*def* client\_ip\_show():

    ip = request.headers.getlist("X-Forwarded-For")[0]

    return "Your IP Address = " + ip

This route is created with debugging in mind. This route was used to develop the client\_ip() helper function. This route basically returns the current user request IP address to verify whether the website is getting access to the IP address or not.

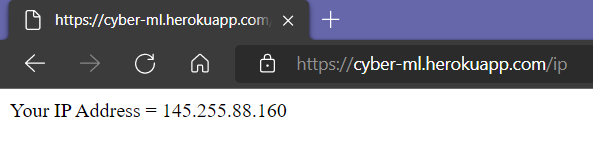


Figure 2 - Get User Request IP.

@app.route("/country")

*def* client\_country\_show():

    ip = client\_ip()

    response = requests.get("http://ip-api.com/json/{}".format(ip))

    js = response.json()

    country = js['countryCode']

    if country:

        return "Your Country is = " + country

    else:

        return "User IP Not Accessible!"

Like the previously mentioned route, this route is also meant for debugging. This route was used to develop the client\_country() helper function. This returns the current user request country of origin to verify if the website has access to the user country of origin or if the online API is working or not.

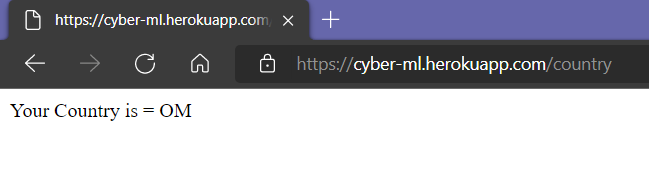


Figure 3 - Get User Request Country of Origin.

@app.route("/getOTP", *methods*=['POST'])

*def* getOTP():

    phone = request.form['phone']

    check = getOTPApi(phone)

    if check:

        return render\_template("enterOTP.html")

This route calls the helper function that delivers the OTP SMS text to the user by taking in the user phone number and passing it to the helper function as an argument. Page routing in Flask makes this route and page secure i.e., the route cannot be accessed without entering a phone number and clicking the form button.

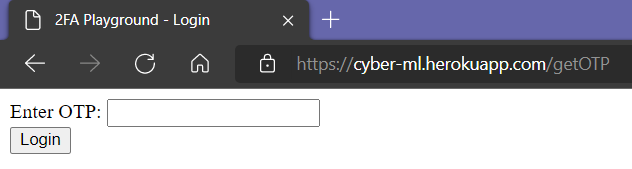


Figure 4 - Send OTP to the User.

@app.route("/validateOTP", *methods*=['POST'])

*def* validateOTP():

    otp = request.form['otp']

    model\_check = model()

    if 'response' in session:

        s = session['response']

        session.pop('response', None)

        if s == otp and model\_check == 'normal':

            return 'Successful Login!'

        else:

            return 'Anomaly IP Detected, Please Retry Login!'

This route receives in the OTP entered by the user.

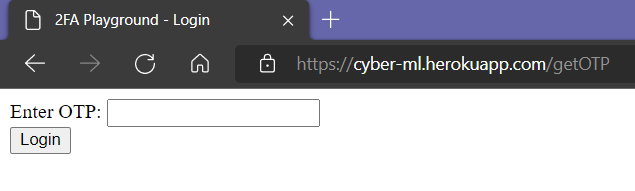


Figure 5 - User enters Received OTP.

It also calls the helper function model() that returns if the user request is normal or anomalistic. By comparing the OTP from the user with the OTP value in the application session along with the predicted result, the following outputs appear:

On correct OTP and normal user request:

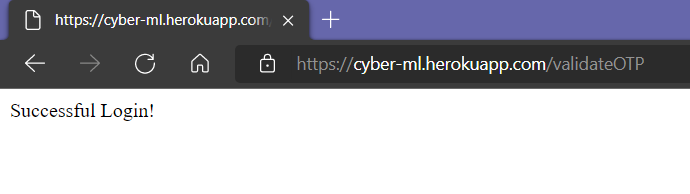


Figure 6 - Correct OTP and Normal User Request.

On either wrong IP or wrong OTP:

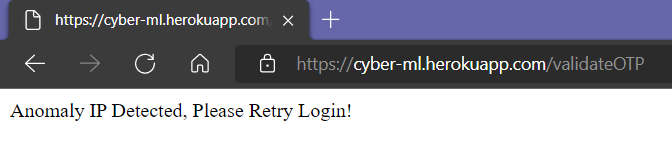


Figure 7 - Either Wrong OTP or Anomalistic User Request.

On the local server, logs will be maintained of each login attempt and the result. The log file is as follows:

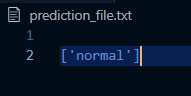


Figure 8 - .txt Log File.

## Machine Learning Model

The Machine Learning Implementation requires further research to finalize it properly. The most crucial part of the implementation of the Machine Learning part is the dataset. The dataset for training the algorithm on anomaly detection in 2FA has to be created since there is none available. Research on the available datasets uncovered a few datasets from which a few features can be extracted and fused together to create a new dummy dataset just for the proof-of-concept. Currently, the dataset employed for this research has only two features, ‘IP’ and ‘Country’.

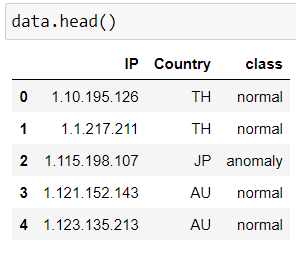


Figure 9 - Sample Training Data.

As it can be seen from the sample dataset, there aren’t many features in the dataset and the dataset is also of the binary classification paradigm. The requirements to digest this dataset therefore are not that high. To this extent, as a proof-of-concept for this research, Logistic Regression will be applied to the dataset. The reasoning behind the selection of Logistic Regression is the fact that it is simple to understand, lightweight and easily implementable. This will help in faster and more efficient development of the complete project.

## Data Preprocessing

Sine the dataset was a fusion of two completely different datasets with no relation in between, there are a few things that need to be done to make the dataset usable and optimal for model digestion. The first thing is to check for null values;

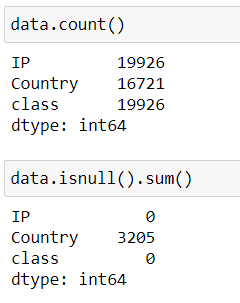


Figure 10 - Total Data Count and Null Value Count.

It can be seen from Figure 10 that the *‘Country’* column contains 3205 null instances out of the total 19926 instances in the complete dataset. Therefore, these null instances need to be dropped as follows;



Figure 11 - Selecting Not Null Instances and Reinitializing the Dataframe.

The code in Figure 11 results in the following;

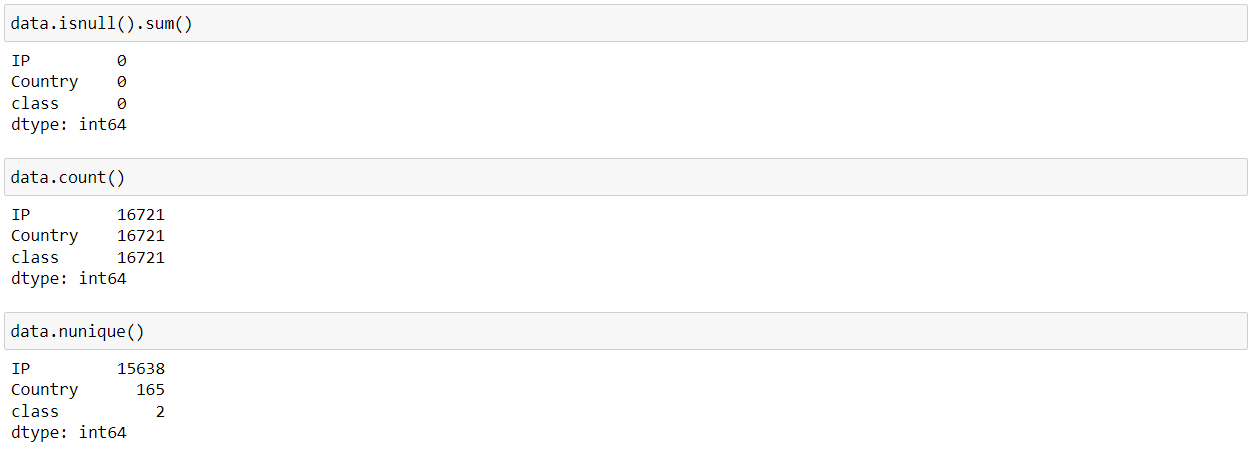


Figure 12 - Null Count, Dataframe Length and Unique Count.

Figure 12 provides more insight into the dataset. The dataset has 15638 unique IP addresses from 165 different countries classified into 2 classes namely **‘normal’** and **‘anomaly’**. Visualizing this will provide a much better understanding of the data distribution based on class.



Figure 13 - Normal Count vs Anomaly Count

The next step in data preprocessing is to perform encoding. It can be seen from Figure 9 that the dataset does not exactly contain integer values rather string values. ML Models cannot digest any other form of data except integer. To make it digestion compatible, encoding has to be done. Scikit-learn library provides simple encoders out of the box. The encoder selected for this research is the Label Encoder. Label Encoding just identifies unique values in a dataset and assigns them an integer value. The reason behind employing Label Encoder is because it is simple to understand, efficient, quick and easy to implement. The result of encoding the dataset is as follows:

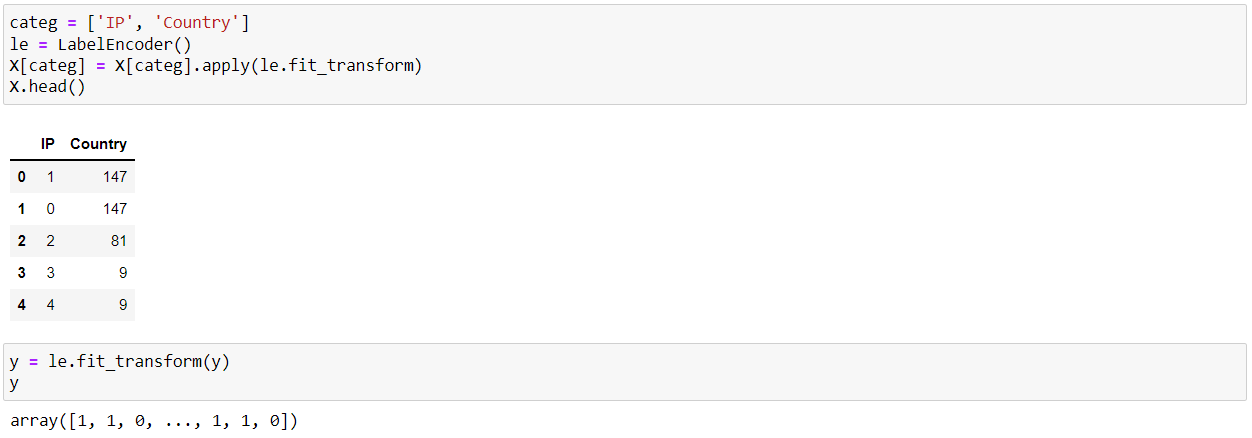


Figure 14 - Label Encoding the Dataset.

It can be observed from Figure 14 that all the values in the dataset have been turned into integers. It can also be seen that the dataset has been separated into 2 variables, **X** being the features of the dataset and **y** being the class output. This is done so that the dataset can be split into testing and training datasets using the tools provided by the scikit-learn library.



Figure 15 - Splitting the Dataset into Testing and Training Sets.

Figure 15 shows the code that splits the dataset into two parts; 75% for training and 25% for testing. With this the basic data preprocessing is complete. Now to fit the Machine Learning Algorithm.

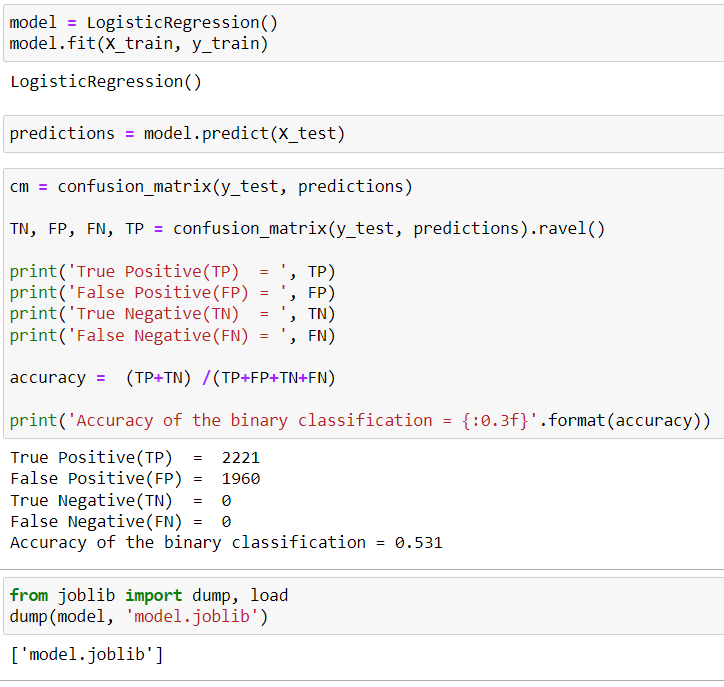


Figure 16 - Model Fitting, Performance Metrics and Model Pickling.

Figure 16 shows three things happening in the process of training the ML model. The first stage is fitting and actually training the model with the training set. The second is to get the algorithm to predict using the testing set to test its performance. Then the results are passed to the confusion matrix, which is a performance metric popularly used for classification problems and the confusion provides a detailed summary of the model’s performance. The accuracy of the model as shown in Figure 16 is 53.1%. This accuracy is not bad but not good either. The next step is to pickle the model. This means to save the model as a file with all its parameters and learning so that the model can be used in different places. This is done so that the model can be used in the 2FA playground.

# Results, Analysis and Evaluation

Figure 16 shows the results the model yielded after training. The results are underwhelming. There are multiple issues behind this. The selection criteria of the algorithm never included the optimization of the parameters and the features or the quality of the dataset. The dataset is a fusion of completely unrelated features from two different datasets, the coefficient of correlation of the dataset was not measured which does not give insight into the correlation between the features of the currently equipped dataset. The aim of this project was to prove that Machine Learning can be used to create a secure verification layer between the user and the 2FA as a 3rd Layer to the existing 2FA model to increase security against phishing and other different attacks. To this extent, the algorithm completely fits within theoretical expectations.

The domain and the topic for this research is a novel approach towards securing the already state-of-the-art 2-factor authentication. Therefore, there isn’t much available regarding datasets for training the algorithm. Due to existing time constraints, creating a proper dataset from scratch was not feasible since the process that needed to be followed for the creation of the dataset was complex and time consuming. The process of dataset creation is as follows:

1. Creating a 2FA playground.
2. Hosting the playground on the internet.
3. Getting people to use the 2FA playground.
4. Recording the IP address, host, location, access port information and other various features from the user request on the website.
5. Encouraging the users to use tools like proxies or VPNs to create user requests on the 2FA playground.
6. Studying the collected dataset to identify data instances of the anomalous nature.
7. Manual annotation of the dataset to implement reinforcement learning.

# Conclusion

The aim of this research project was to show that Machine Learning can be used as a powerful tool in increasing online user security against phishing and other attacks of the same nature. To this extent, the objectives of this research were to create a 2FA playground that the user can interact with. The aim of this 2FA playground was to mimic real-time 2FA. This playground was then integrated with the Machine Learning model created and trained to identify IP addresses anomalous in nature.

The algorithm used to develop the model was Logistic Regression. The reason behind it was that the dataset defined the paradigm of Machine Learning as Binary Classification. Logistic Regression is one of the simpler algorithms that can be easily implemented for Binary Classification. It is fast, efficient and easy to understand. This project is a proof-of-concept for other researchers that pursue research in the same or similar domain.

The performance metrics of the model yielded underwhelming results. This is because the dataset was a rough fusion of different features from two different datasets. Although data preprocessing was done, it was not enough to get the results up to above 80%. Although the results were not satisfactory, the functioning of the application was correct and working as expected.

# Recommendations

There is a lot that can be improved in this research. The project was limited by its very virtue of being a novel approach towards internet security. There were no existing datasets that were strong or proper enough to get the results needed from the ML model. The dataset has to be self-created for this research. The process of dataset creation however is discussed in detail in the [Results](#_Results,_Analysis_and) subsection.

The second improvement will be in the data preprocessing section. Due to the weakness of the dataset used in this project, proper high-end data preprocessing techniques like correlation matrices, PCA and feature selection were not applied. These techniques can be used to significantly improve the strength of the dataset that will result in better ML model performance.

The third improvement will be the algorithm selection process. Once a proper preprocessed dataset is available, tools like Grid Search from scikit-learn library can be used to automatically select the perfect classification algorithm with hyper-tuned parameters to provide the best results. If the dataset allows it, deep learning can also be used for this scenario.

The fourth improvement can be done to the 2FA playground. The current playground is just the user entering his phone number to receive the OTP and entering it in another form to verify the integrity of the user logging in. Although this works perfectly according to the aims of this research project, the design and the overall GUI can be improved more to appeal to the public and collect data easier. It can also be better to apply it to a real-life working 2FA model to get even better results.`

# References

* Irvan, M., Thao, T., Kobayashi, R., Nakata, T. and Yamaguchi, R., 2021. Learning from Smartphone Location Data as Anomaly Detection for Behavioral Authentication through Deep Neuro-evolution. Seventh International Conference on Information Systems Security and Privacy.
* Ali, G., Ally Dida, M. and Elikana Sam, A., 2020. Two-Factor Authentication Scheme for Mobile Money: A Review of Threat Models and Countermeasures. Future Internet, 12(10), p.160.
* Anderson, W. and Simske, S., 2020. At-Home Healthcare Through Smart-Environmental Sensing, Including Biometrics for Multi-Factor Authentication. 2020 IEEE International Conference on Healthcare Informatics (ICHI).
* Asani, E., Longe, O., Balla, A., Ogundokun, R. and Adeniyi, E., 2020. Secure Human-Computer Interaction. Handbook of Research on the Role of Human Factors in IT Project Management, pp.149-163.
* Iyer, A., Karthikeyan, D., Hasan Khan, M. and Binu, D., 2020. AN ANALYSIS OF ARTIFICIAL INTELLIGENCE IN BIOMETRICS-THE NEXT LEVEL OF SECURITY. Journal of critical reviews, 7(1).
* Karim Abdul-Hassan, A. and Hasson Hadi, I., 2019. INTELLIGENT AUTHENTICATION FOR IDENTITY AND ACCESS MANAGEMENT: A REVIEW PAPER. Iraqi Journal for Computers and informatics, 45(1), pp.6-10.
* Kesem, L., 2021. AI Hand Pattern Authentication Method. Boise State University.
* Kightlinger, D., 2019. Beyond 2FA: Secure Your Critical Assets with Risk-Based Multifactor Authentication. [Online] Security Intelligence. Available at: <https://securityintelligence.com/posts/beyond-2fa-secure-your-critical-assets-with-risk-based-multifactor-authentication/> [Accessed 5 April 2022].
* Kwon, B., Sharma, P. and Park, J., 2019. CCTV-Based Multi-Factor Authentication System. Journal of Information Processing Systems, 15(4), pp.904-919.
* Townsend, A., 2021. AI's Role in Authentication. [Online] OneLogin. Available at: <https://www.onelogin.com/blog/ai-authentication> [Accessed 5 April 2022].
* Boud, D., Keogh, R. and Walker, D., 1985. Reflection - Turning Experience into Learning. 1st ed. Routledge.
* Boyd, E. and Fales, A., 1983. Reflective Learning. Journal of Humanistic Psychology, 23(2), pp.99-117.
* Gibbs, G., 1988. Learning by doing. 1st ed. [London]: FEU.
* Jarvis, P., 1992. Reflective practice and nursing. Nurse Education Today, 12(3), pp.174-181.
* Mezirow, J., 1981. A Critical Theory of Adult Learning and Education. Adult Education, 32(1), pp.3-24.