An Industrial Oriented Mini Project (CS704PC)

on

"PREDICTION AND MODELLING OF CYBER HACKING DATA BREACHES"

Submitted

in the partial fulfilment of the requirements for

the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

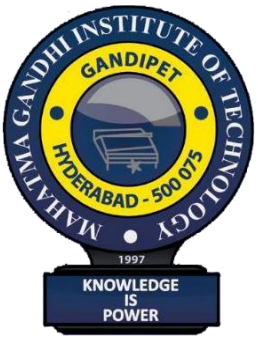
MR. CHIKKALA P S M V S N NAIDU

(18261A0512)

Under the guidance of

DR. M RAMA BAI

(Professor)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

Kokapet (V), Gandipet (M), Hyderabad.

Telangana -500 075. (India)

2021-2022

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)
GANDIPET, HYDERABAD – 500 075. Telangana

CERTIFICATE

This is to certify that the Industrial Oriented Mini Project entitled "**Prediction and Modelling of Cyber Hacking Data Breaches**" submitted by **Chikkala P S M V S N Naidu** bearing **Roll No: 18261A0512** in partial fulfillment for the award of **B. Tech** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** is a record of bonafide work carried out under the supervision of **Dr. M. Rama Bai, Professor, Department of CSE.**

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

Project Guide	Head of the Department
Dr. M. Rama Bai	**Dr. C. R. K. Reddy**
Professor, Dept. of CSE	Professor, Dept. of CSE

External Examiner

DECLARATION

This is to certify that the work reported in this project titled "**PREDICTION AND MODELLING OF CYBER HACKING DATA BREACHES"** is a record of work done by me in the Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad.

No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text. The report is based on the work done entirely by me and not copied from any other source.

CHIKKALA P S M V S N NAIDU

(18261A0512)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success. They have been a guiding light and source of inspiration towards the completion of the project.

I would like to express my sincere thanks to **Dr. K. Jaya Sankar, Principal MGIT**, for providing the work facilities in the college.

I am also thankful to **Dr. C.R.K Reddy, Professor & Head of Department**, Dept. of Computer Science and Engineering for providing excellent infrastructure and a conducive atmosphere for completing this project successfully.

I am also extremely thankful to my Project Coordinators, **Dr. C.R.K Reddy, Professor and Head of Department, Dr. B Prasanthi, Senior Assistant Professor, and Ms. D Deepika, Assistant Professor,** for their valuable suggestions and interest throughout the course of this project.

I would like to express my sincere gratitude to my Project Guide, **Dr. M. Rama Bai, Professor,** who has supported me throughout the project with patience and knowledge.

I convey my heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed.

Finally, I would like to take this opportunity to thank my family for their support throughout the work. I sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

CHIKKALA P S M V S N NAIDU
(18261A0512)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Analyzing cyber incident datasets is an important method for deepening our understanding of the evolution of the threat situation. This is a relatively new research topic, and many studies remain to be done. In this proposed work, statistical analysis of a breach incident dataset corresponding to 14 years (2007-2021) of cyber hacking breaches that include different type of attacks is analyzed. This project show that, in contrast to the findings reported in the literature, both the hacking breach incident inter-arrival times and the breach sizes should be modelled by stochastic processes, rather than by distributions because they exhibit autocorrelations. So, proposing stochastic process models to respectively fit the inter-arrival times of breaches. This stochastic process models forecasts the probability of various outcomes under different conditions, using random variables. It also shows that these models can predict the inter-arrival times and the breach sizes.

To get deeper insights into the evolution of hacking breach incidents, both qualitative and quantitative trend analysis is conducted on the dataset using Arima model. This trend analysis is taken as a major factor for prediction of the breaches. Testing is done on the model by using various error metrics to improve the results.

1. INTRODUCTION

A Cyber Hacking Data Breach is a confirmed incident in which sensitive, confidential data has been accessed or disclosed in an unauthorized fashion. Data breaches are one of the most devastating cyber incidents. These incidents range from concerted attacks by individuals who hack for personal gain, organized crime, political activists, or national governments, to poorly configured system security or careless disposalof used computer equipment or data storage media.

Data breaches may involve financial information such as credit card and debit card details, bank details, personal health information (PHI), Personally identifiable information (PII), trade secrets of corporations or intellectual property. Data breaches may involve overexposed and vulnerable unstructured data – files, documents, and sensitive information.

While technological solutions can harden cyber systems against attacks, data breaches continue to be a big problem. This motivates us to characterize the evolution of data breach incidents. This not only will deep our understanding of data breaches, but also shed light on other approaches for mitigating. The Identity Theft Resource Center and Cyber Scout reports 1,093 data breach incidents in 2018, which is 40% higher than the 780 data breach incidents in 2017.

Data breaches expose 4.1 billion records in first six month of 2019.The first six months of 2019 have seen more than 3800 publicly disclosed breaches exposing an incredible 4.1 billion compromised records. In 2019, the number of data breaches amounted to 1,473 with over 164.68 million sensitive records exposed. Data breaches have gained attention with the increasing use of digital files and companies and users large reliance on digital data. In 2020 at least 7.9 billion records, including credit card numbers, home addresses, phone numbers and other highly sensitive information, have been exposed through data breaches.

1.1 Problem Definition

Data breaches are one of the most devastating cyber incidents. To create a model that can predict the number of breaches occurred due to cyber hacking. This model is a time series model which can be useful to know when the breach has occurred, and number of records lostat a particular time. This helps them to recover or take remedial actions on their data security system. In proposed work the model analyzes previous data and produce the prediction. This prediction is plotted as graph using various modules and the result can be observed.

1.2 Existing System

Existing systems analyzed a dataset from the point of view of actuarial modelling and pricing. Authors used a variant of the Gompertz model to analyze the growth of computer and Internet-related crimes. Using datasets collected at a honeypot, authors exploited their statistical properties including long-range dependence and extreme values to describe and predict the number of attacks against the honeypot.

The present study is motivated by several questions that have not been investigated until now, such as: data breaches caused by cyber-attacks increasing, decreasing, or stabilizing? A principled answer to this question will give us a clear insight into the overall situation of cyber threats. This question was not answered by previous studies.

1.3 Proposed System

The proposed work will show that both the hacking breach incident interarrival times (reflecting incident frequency) and breach sizes which should be modelled by stochastic processes, rather than by distributions. It will find that a particular point process can adequately describe the evolution of the hacking breach incidents inter-arrival time and Time Series Analysis based (ARIMA model) is used to analyze and predict cyber hacking breaches.

1.4 Requirements Specification

1.4.1 Hardware Requirements

- Processor : Intel core (i5 or higher)
- RAM : 4 GB
- ROM : 8 GB

1.4.2 Software Requirements

- Operating System : Windows/ MacOS
- Programming Language : Python
- Integrated Development Environment (IDE) : Jupyter Notebook

2. LITERATURE SURVEY

1. **R.Raja Subramanian; Rithvik Avula; Paramkusam Sree Surya; "Modeling and Predicting Cyber Hacking Breaches", IEEE May 2021. [1]**

   The primary aim of this research work is to create a machine learning model, which trains in Realtime and monitors the website or a system and trains from the state-of-art attacks. The proposed model has created a web application using Django, which takes the data from multiple sources such as Amazon, Flipkart, Snapdeal, and Shop clues, which shows the data that is safe to obtain from the website. Then, the data will be sorted on our page and then it will be made secured and illegal for the external people to access the data from our website and the proposed model will monitor the website 24/7.

2. **Zijian Fang; Maochao Xu; Shouhuai Xu; Taizhong Hu; "A Framework for Predicting DataBreach Risk: Leveraging Dependence to Cope with Sparsity" IEEE Jan 2021.[2]**

   In this paper, they initiated the study of modeling and predicting risk in enterprise-level data breaches. This problem is challenging because of the sparsity of breaches experienced by individual enterprises over time, which immediately disqualifies standard statistical models because there are not enough data to train such models. As a first step towards tackling the problem, it proposes an innovative statistical framework to leverage the dependence between multiple time series. To validate the framework, we apply it to a dataset of enterprise-level breach incidents. Experimental results show its effectiveness in modeling and predicting enterprise-level breach incidents.

3. **Somchart Fugkeaw; Nichakorn Kuasomboon; Pathitta Panakitkul; "DBIM: An Efficient and Resilient PII Data Breach Management System" IEEE May 2021.[3]**

   This paper presents the design and development of an automated PII data breach incident management system as a complimentary tool for privacy management for PDPA compliance. DBIM possesses three key functions including incident classification based on incident impact analysis and support vector machine (SVM), response plan generation based on rule-based system, and incident follow-up. With the proposed SVM and rule-based approach, the proposed system is efficient and resilient in handling a high volume of incident cases. To substantiate the efficiency of proposed system, they developed a system prototype to validate functionality of the system.

4. **Dmitry Zegzhda; Daria Lavrova; Aleksei; "Detection of information security breaches in distributed control systems based on values prediction of multidimensional time series" IEEE May 2019. [4]**

   Proposed an approach for information security breaches detection in distributed control systems based on prediction of multidimensional time series formed of sensor and actuator data by using CNN model for multidimensional time series prediction, which allows us to detect information security breaches.

5. **Nikita Tresa Cyriac; Lipsa Sadath; "Is Cyber Security Enough- A study on Big Data Security Breaches in Financial Institutions" IEEE Nov 2019. [5]**

   The proposed work presents MECA (Model to Encounter Cyber Attacks) which could be adopted by financial institutions. MECA proposes the intelligent application of IDS, Security Patches and Big Data Analytics to mitigate cyber challenges in financial institutions. The focus of MECA is to technologically orient institutes to handle dynamic cyber breaches, to train employees to effectively apply this intelligent application and to develop them as the first line of defense to cyber-attacks. The paper also investigates the actors in a cyber-attack and the tools predominantly used to reach their objective. It casts light over the general cyber-attack framework, its stages, and the effect it has on the financial institution system.

6. **Yong Fang; Yusong Guo; Cheng Huang; And Liang Liu; "Analyzing and Identifying Data Breaches in Underground Forums" IEEE Apr 2019. [6]**

   Recently, underground forums played a crucial role in trading and exchanging leaked personal information. Meanwhile, the forums have been gradually used as data breaches information sources. Therefore, it shows an upward trend in announcing the results of data theft by posting in the forums. Identifying these threads can make the compromised third-party respond quickly to the data breach incident. For this purpose, they presented a system to identify the threads which are related to data breaches automatically. The system can monitor and discover data breaches in underground forums in real-time. They compared various supervised classification algorithms in this application scenario and selected the best method for the classifier.

Table 2.1: Literature survey for Prediction and Modelling of Cyber Hacking Data Breaches

S.No	Year	Name of the Author	Title of paper	Method	Advantages	Disadvantages
1.	2021	R.Raja Subramanian, Rithvik Avula. Paramkusam Sree Surya, International Conference on Intelligent Computing and Control Systems, IEEE, 2021	Modelling and Predicting Cyber Hacking Breaches	The proposed model has created a web application using Django	It can forecast the breach situation and can put out the most accurate statistical analysis in the field.	It does not predict breaches but shows only statistics
2.	2021	Zijian Fang, Maochao Xu , Shouhuai Xu, and Taizhong Hu, IEEE Transactions on Information Forensics and Security, Volume: 16, IEEE, 2021	A Framework for Predicting Data Breach Risk: Leveraging Dependence to Cope with Sparsity	Gompertz model is used to analyze the growth of computer-related crimes	This framework analyzes a breach dataset and showed that the statistical distribution of breach sizes.	It does not predict when the next breach will occur to an enterprise
3.	2021	Somchart Fugkeaw, Nichakorn Kuasomboon, Pathitta Panakitkul, 13th International Conference on Knowledge and Smart Technology, IEEE, 2021	DBIM: An Efficient and Resilient PII Data Breach Incident Management System	Support Vector Machine (SVM) Response plan generation on rule-based system	SVM and rule-based approach the system is efficient and resilient in handling a high volume of incident cases.	SVM require very high computational power

4.	2019	Dmitry Zegzhda, Daria Lavrova, Aleksei Khushkeev, 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), IEEE, 2019	Detection of information security breaches in distributed control systems based on values prediction of multidimensional time series	CNN model for multidimension al time series prediction, which allows us to detect information security breaches.	Optimal result has been achieved in detecting cyberattacks by predicting using the CNN neural networks.	Convoluti onal Neural Networks takesa lot of time to train the data.
5.	2019	Nikita Tresa Cyriac; Lipsa Sadath, 2019 4th International Conference on Information Systems and Computer Networks, IEEE, 2019	Is Cyber Security Enough- A study on Big Data Security Breaches in Financial Institutions	MECA (Model to Encounter Cyber Attacks) which focuses on developing anintelligent application.	Financial Institutions on adopting MECA can ensure the security of their data and financial Resources.	MECA requiresto develop an innovative security framework thatmeets the technical requireme nts to handle cyber-attacks.
6.	2019	Yong Fang; Yusong Guo; Cheng Huang; And Liang Liu, IEEE Access, Volume: 7, IEEE, 2019	Analyzing and Identifying Data Breaches in Underground Forums	Supervised classification algorithms are used.	Analyzing and identifying these threads makes compromised third parties understand their damage early and respond to incidents in time	Identificati on of breaches is not accurate.

3.METHODOLOGY FOR PREDICTION AND MODELLING OF CYBER HACKING DATA BREACHES

3.1 Architecture

The system basically uses Jupyter notebook which sets the path to read the data set. The data set is revived from the data base and send for data pre-processing. In the pre-processing the data isextracted, selected values are converted to continuous values then the data is used to train model. After training evaluate new data with the model to get predicted values. Figure 3.1 shows the system architecture of the proposed system.

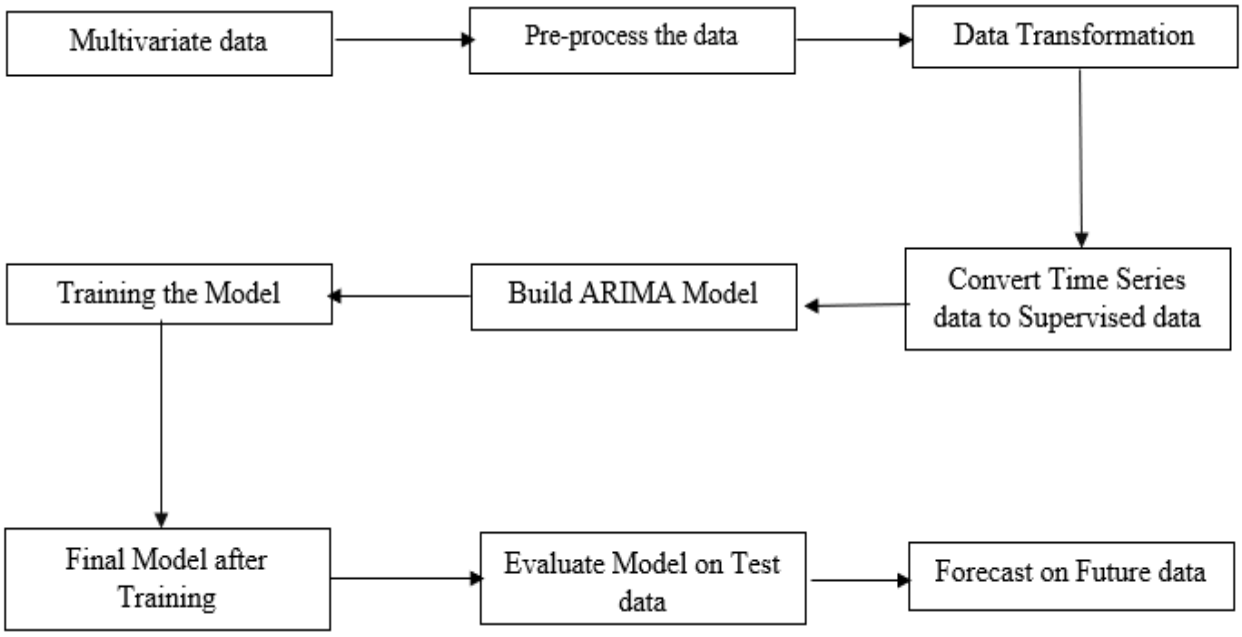


Fig. 3.1. General Training Workflow

3.1.1 Dataset description

The hacking breach dataset analyzed in this proposed work is obtained from the Privacy Rights Clearinghouse (PRC),[7] which is the largest and most extensive dataset that is shown in fig. 3.2. Since we focus on hacking breaches, we disregard the negligent breaches and the other sub-categories of malicious breaches (i.e., insider, payment card fraud, and unknown). From the remaining raw hacking breaches data, it is further disregarding the incomplete records with unknown/missing hacking breach sizes because breach size is one of the objects for this

study. The resulting dataset in fig 3.2 contains 600 hacking breach incidents in the United States between January 1st, 2007, and October 30th, 2019. The hacking breach victims span over 7 industries: businesses-financial and insurance services (BSF); businesses retail/merchant including online retail (BSR); businesses-other (BSO); educational institutions (EDU); government and military (GOV); healthcare, medical providers and medical insurance services (MED); and nonprofit organizations (NGO).[1] The dataset is represented by a sequence, denoted by $\{(t_i , y_{ti})\}_{0 \le i \le 600}$, where ti represents the day on which there is an incident of breach size yti (i.e., the number of private data records thatare breached by the incident), and t0 is the day on which observation starts (i.e., t0 does notcorrespond to the occurrence of any incident).

The inter-arrival times are $d_i = t_i - t_{i-1}$, where $i = 1, 2, \ldots, 600$. Among the ti's, most days have onesingle incident report, 52 days with 2 incidents on each day, 7 days with 3 incidents on each day, and one day (02/26/2016) with 7 incidents. We caution that the dataset does not necessarily contain all the hacking breach incidents, because there may be unreported ones. Moreover, the dates corresponding to the incidents are the days on which the incidents are reported, rather than the dates on which the incidents took place. Nevertheless, this dataset (or data source) represents the best dataset that can be obtained in the public domain. Therefore, analysis of it will shed light on the severeness of the data breach risk, and the analysis methodologies can be adopted or adapted to analyze more accurate datasets of this kind when they become available in the future.

```
data=pd.read_csv("C:/Users/naras/Downloads/cyber_breach_data.csv")
```

```
data.head()
```

	Date_Made_Public	Company	City	State	Type_of_breach	Type_of_organization	Total_Records	Description_of_incident	Information_Source
0	01-10-2007	George Mason University	Fairfax	Virginia	HACK	EDU	32,000	Names, photos, and Social Security numbers of ...	Dataloss DB
1	1-18-2007	University of California, San Diego	San Diego	California	HACK	EDU	3,500	A hacker breached the security of two Universi...	Dataloss DB
2	1-22-2007	University of Northern Colorado	Greeley	Colorado	PORT	EDU	15,790	A hard drive was lost or stolen. It contained ...	Dataloss DB
3	02-12-2007	Science Applications International Corp. (SAIC)	San Diego	California	STAT	BSO	45,000	On January 25 thieves broke \n into...	Dataloss DB
4	2-15-2007	ChoicePoint	Alpharetta	Georgia	INSD	BSO	1,63,000	Fraudsters who presented themselves as legitim...	Security Breach Letter

Fig. 3.2. Display of first 5 rows in dataset

3.1.2 Preprocessing

Because of observation, as mentioned above, some days have multiple hacking breach incidents, one may suggest treating such multiple incidents as a single “combined” incident (i.e., adding their number of breached records together). However, this method is not sound because the multiple incidents may happen to different victims that have different cyber systems. Given that the time resolution of the dataset is a day, multiple incidents that are reported on the same data may be reported at different points in time of the same day (e.g., 8pm vs. 10pm). It proposes generating small random time intervals to separate the incidents corresponding to the same day. Specifically, it randomly orders the incidents corresponding to the same day, and then insert a small- and random- time interval in between two consecutive incidents (for the first interval, the starting point is midnight), while assuring that these incidents correspond to the same day (e.g., the two incidents on a two-incident day may be assigned at 8am and 1pm).

3.1.3 Data Transformation

The goal of the data transformation process is to extract data from a source, convert it into a usable format, and deliver it to a destination. This entire process is known as ETL (Extract, Load, Transform). During the extraction phase, data is identified and pulled from many different locationsor sources into a single repository.

Data extracted from the source location is often raw and not usable in its original form. To overcome this obstacle, the data must be transformed. This is the step in the ETL process that adds the most value to your data by enabling it to be mined for business intelligence. During transformation, a number of steps are taken to convert it into the desired format. In some cases, data must first be cleansed before it can be transformed. Data cleansing prepares the data for transformation by resolving inconsistencies or missing values.

3.1.4 Convert Time series data to Supervised data

Time series data can be phrased as supervised learning. Given a sequence of numbers for a time series dataset, we can restructure the data to look like a supervised learning problem. We can do this by using previous time steps as input variables and use the next time step as the output variable. We can restructure this time series dataset as a supervised learning problem by using the value at the previous time step to predict the value at the next time-step as shown in fig 3.3.

```
y = time_series['Total_Records'].resample('MS').mean()
```

```
y.plot(figsize=(15, 6))
plt.show()
```

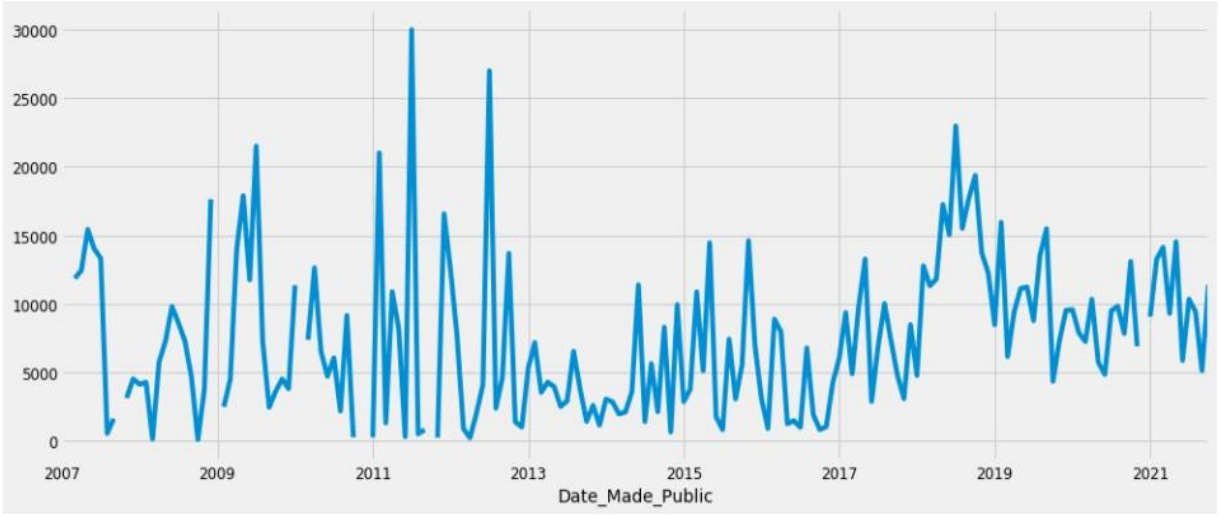


Fig. 3.3 Time series data represented in Supervised data

3.1.5 ARIMA Model

ARIMA stands for Auto-Regressive Integrated Moving Average. It is actually a class of models that 'explains' a given time series [8] based on its own past values, that is, its own lags and the laggedforecast errors, so that equation can be used to forecast future values.

There are three integers (p, d, q) that are used to parametrize ARIMA models.

p is the number of autoregressive terms (AR part). It allows to incorporate the effect of past values into our model. Intuitively, this would be similar to stating that it is likely to be warm tomorrow if it has been warm since the past 3 days.

d is the number of nonseasonal differences needed for stationarity. Intuitively, this would be similar to stating that it is likely to be same temperature tomorrow if the difference in temperature in the last three days has been very small.

q is the number of lagged forecast errors in the prediction equation (MA part). This allows us to set the error of our model as a linear combination of the error values observed at previous time points in the past.

Mathematical Explanation of ARIMA Model:

The acronym ARIMA stands for Auto-Regressive Integrated Moving Average. Lags of the stationarized series in the forecasting equation are called "autoregressive" terms, lags of the forecast errors are called "moving average" terms, and a time series which needs to be differenced to be made stationary is said to be an "integrated" version of a stationary series.

A nonseasonal ARIMA model is classified as an "ARIMA (p, d, q)" model, where:

- **p** is the number of autoregressive terms,
- **d** is the number of nonseasonal differences needed for stationarity, and
- **q** is the number of lagged forecast errors in the prediction equation.

The forecasting equation is constructed as follows. First, let y denote the d^{th} difference of Y, which means:

If d=0: $y_t = Y_t$

If d=1: $y_t = Y_t - Y_{t-1}$

If d=2: $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

Note that the second difference of Y (the d=2 case) is not the difference from 2 periods ago. Rather, it is the *first-difference-of-the-first difference*, which is the discrete analog of a second derivative, i.e.,the local acceleration of the series rather than its local trend.

In terms of y, the general forecasting equation is:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q} \quad \ldots\ldots(1)$$

Here the moving average parameters (θ's) are defined so that their signs are negative in the equation, following the convention introduced by Box and Jenkins. Some authors and software (including the R programming language) define them so that they have plus signs instead. When actual numbers are plugged into the equation, there is no ambiguity, but it's important to know which convention your software uses when you are reading the output. Often the parameters are denoted there by AR(1), AR(2), …, and MA(1), MA(2), … etc..

To identify the appropriate ARIMA model for Y, you begin by determining the order of differencing

(d) needing to stationarize the series and remove the gross features of seasonality, perhaps in conjunction with a variance-stabilizing transformation such as logging or deflating. If you stop at this point and predict that the differenced series is constant, you have merely fitted a random walk or random trend model. However, the stationarized series may still have autocorrelated errors, suggesting that some number of AR terms ($p \geq 1$) and/or some number MA terms ($q \geq 1$) are also needed in the forecasting equation.

The process of determining the values of p, d, and q that are best for a given time series will be discussed in later sections of the notes (whose links are at the top of this page), but a preview of some of the types of *nonseasonal* ARIMA models that are commonly encountered is given below.

ARIMA(1,0,0) = first-order autoregressive model: if the series is stationary and autocorrelated, perhaps it can be predicted as a multiple of its own previous value, plus a constant. The forecasting equation in this case is

$$\hat{Y}_t = \mu + \phi_1 Y_{t-1} \quad \text{.....(2)}$$

which is Y regressed on itself lagged by one period. This is an "ARIMA(1,0,0)+constant" model. If the mean of Y is zero, then the constant term would not be included. [9]

If the slope coefficient ϕ_1 is positive and less than 1 in magnitude (it *must* be less than 1 in magnitude if Y is stationary), the model describes mean-reverting behavior in which next period's value should be predicted to be ϕ_1 times as far away from the mean as this period's value. If ϕ_1 is negative, it predicts mean-reverting behavior with alternation of signs, i.e., it also predicts that Y will be below the mean next period if it is above the mean this period.

In a *second-order* autoregressive model (ARIMA(2,0,0)), there would be a Y_{t-2} term on the right as well, and so on. Depending on the signs and magnitudes of the coefficients, an ARIMA(2,0,0) model could describe a system whose mean reversion takes place in a *sinusoidally oscillating* fashion, like the motion of a mass on a spring that is subjected to random shocks.

ARIMA(0,1,0) = random walk: If the series Y is not stationary, the simplest possible model for it isa random walk model, which can be considered as a limiting case of an AR(1) model in which the autoregressive coefficient is equal to 1, i.e., a series with infinitely slow mean reversion[10]. Theprediction equation for this model can be written as:

$$\hat{Y}_t - Y_{t-1} = \mu \text{ **or** equivalently } \hat{Y}_t = \mu + Y_{t-1} \quad \text{.....(3)}$$

where the constant term is the average period-to-period change (i.e. the long-term drift) in Y. This model could be fitted as a *no-intercept regression model* in which the first difference of Y is the dependent variable. Since it includes (only) a nonseasonal difference and a constant term, it is classified as an "ARIMA(0,1,0) model with constant." The random-walk-without-

drift model would be an ARIMA(0,1,0) model *without* constant.

3.1.1 Training the model using SARIMAX Function

SARIMAX (Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors) is an updated version of the ARIMA model. ARIMA includes an autoregressive integrated moving average, while SARIMAX[11] includes seasonal effects and eXogenous factors with the autoregressive and moving average component in the model. Therefore, we can say SARIMAX is a seasonal equivalent model like SARIMA and Auto ARIMA.

Another seasonal equivalent model holds the seasonal pattern; it can also deal with external effects. This feature of the model differs from other models. For example, in a time series, the temperature has seasonal effects like it is low in winter, high in summers. Still, with the effect of external factors like humidity, the temperature in winter is increased and due to rain, there is a chance of lower temperature. We can't predict the exact value for these factors if they do not appear in a cyclic or any seasonal behavior. Other models are not capable of dealing with this kind of data.

In the SARIMAX models parameter as shown in fig 3.4, we need to provide two kinds of orders. The first one is like the ARIMAX model (p, d, q), and the other is to specify the effect of the seasonality; we call this order a seasonal order in which we are required to provide four numbers.

(Seasonal AR specification, Seasonal Integration order, Seasonal MA, Seasonal periodicity)

```
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]

print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```

```
Examples of parameter combinations for Seasonal ARIMA...
SARIMAX: (0, 0, 1) x (0, 0, 1, 12)
SARIMAX: (0, 0, 1) x (0, 1, 0, 12)
SARIMAX: (0, 1, 0) x (0, 1, 1, 12)
SARIMAX: (0, 1, 0) x (1, 0, 0, 12)
```

Fig. 3.4 SARIMAX Function Combinations

3.2 Required Libraries in proposed work

3.2.1 NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is open- source software andhas many contributors.

3.2.2 Scikit-learn

Scikit-learn is a free software machine learning library for the Python programming language. This library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. We use Scikit-learn for converting between RGB and LAB color spaces.

3.2.3 Pandas

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the NumPy package, and its key data structure is called the Data Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables.

3.2.4 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.

3.2.5 Seaborn

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with panda's data structures. Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

3.2.6 Imblearn

In machine learning, while building a classification model we sometimes come to situations where we do not have an equal proportion of classes. This is called a class imbalance. Imblearn techniques help to either up sample the minority class or down sample the majority class to match the equal proportion.

3.2.7 Warning

Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn't warrant raising an exception and terminating the program. For example, one might want to issue a warning when a program uses an obsolete module.

3.2.8 Statsmodels

statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics is available for each estimator. The results are tested against existing statistical packages to ensure that they are correct.

3.3 Diagrammatic Representation

UML Diagrams

The Unified Modelling Language (UML) is a standard language for writing software blueprints. The UML is a language for:

- **Specifying**: It is just like a blueprint created by an architect prior to the construction.
- **Visualizing**: Visualizing is concerned with deep analysis of system to be constructed.
- **Constructing**: Modelling also provide us mechanism which are essential while constructing a system.
- **Documenting**: Finally, modelling justifies its importance by applying all its credentials to be bounded in a piece of paper referred as document.

The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modelling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modelling yields an understanding of a system.

3.3.1 Use Case Diagram

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a systemis analyzed together its functionalities use cases are prepared and actors are identified.

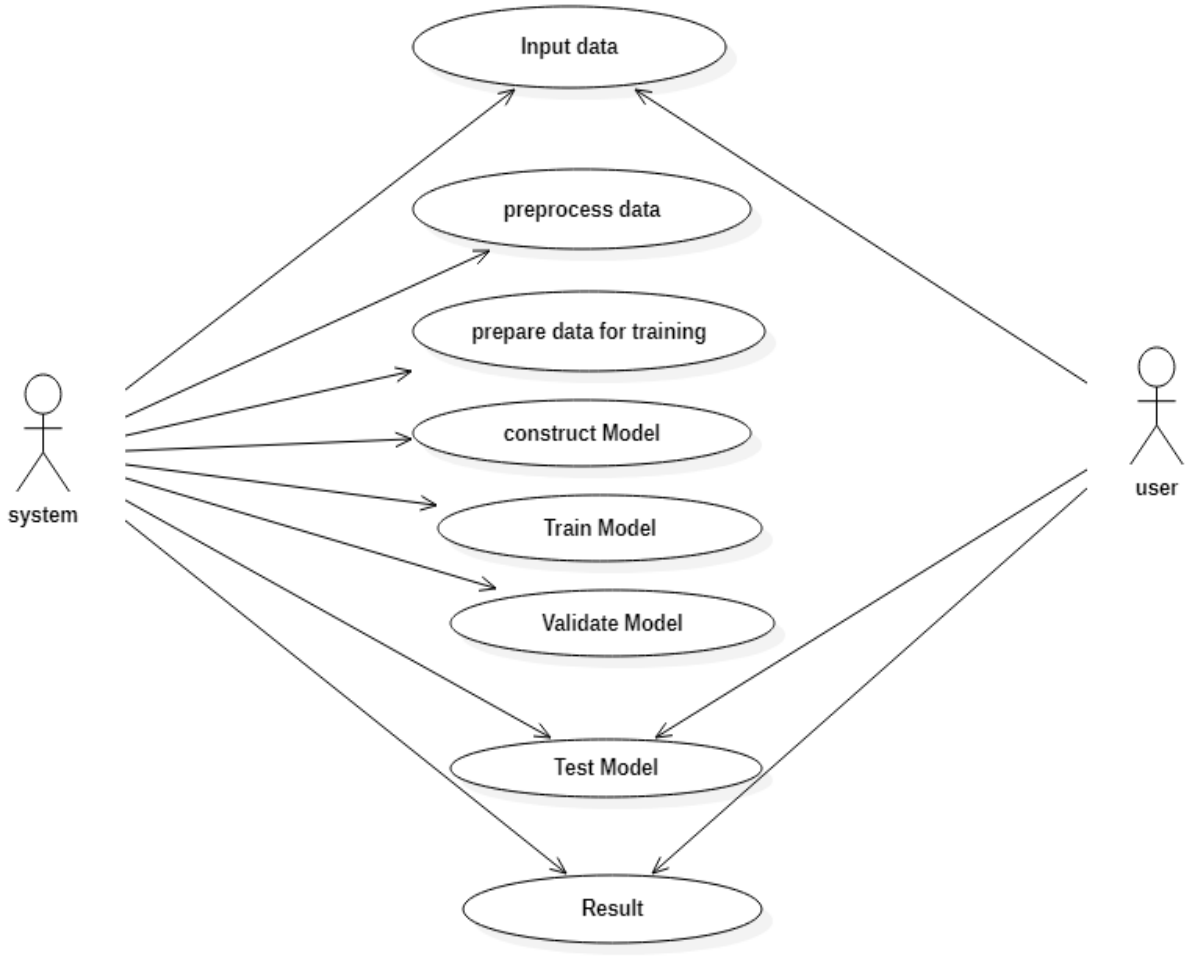


Fig. 3.5. Behaviour model of the system

In Figure 3.5, the user loads the dataset and test the model. While the system applies preprocessing techniques to clean data, builds a ARIMA model, trains and tests the model to generate results.

3.3.2 Activity Diagram

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another**.**

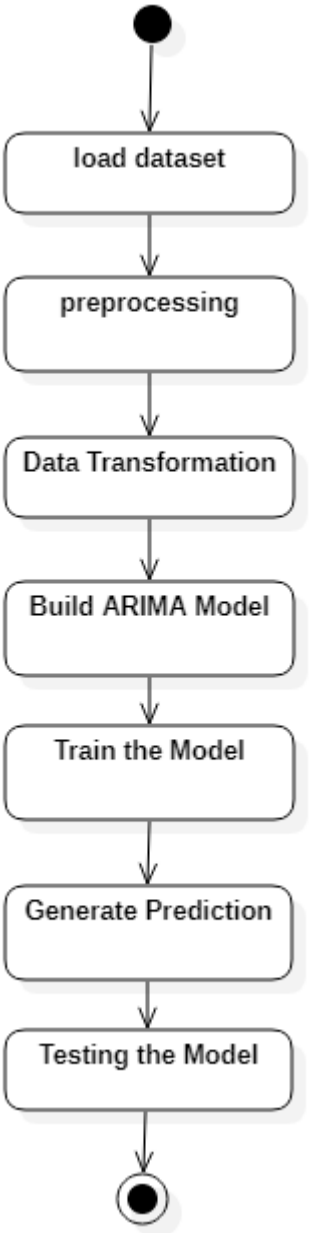


Fig. 3.6. Flow of different activities involved in the proposed work

The flow of control in this project can be observed in the Figure 3.6. Where a dataset is first loaded and after preprocessing, generating a model, training, and testing it displays results.

3.3.3 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

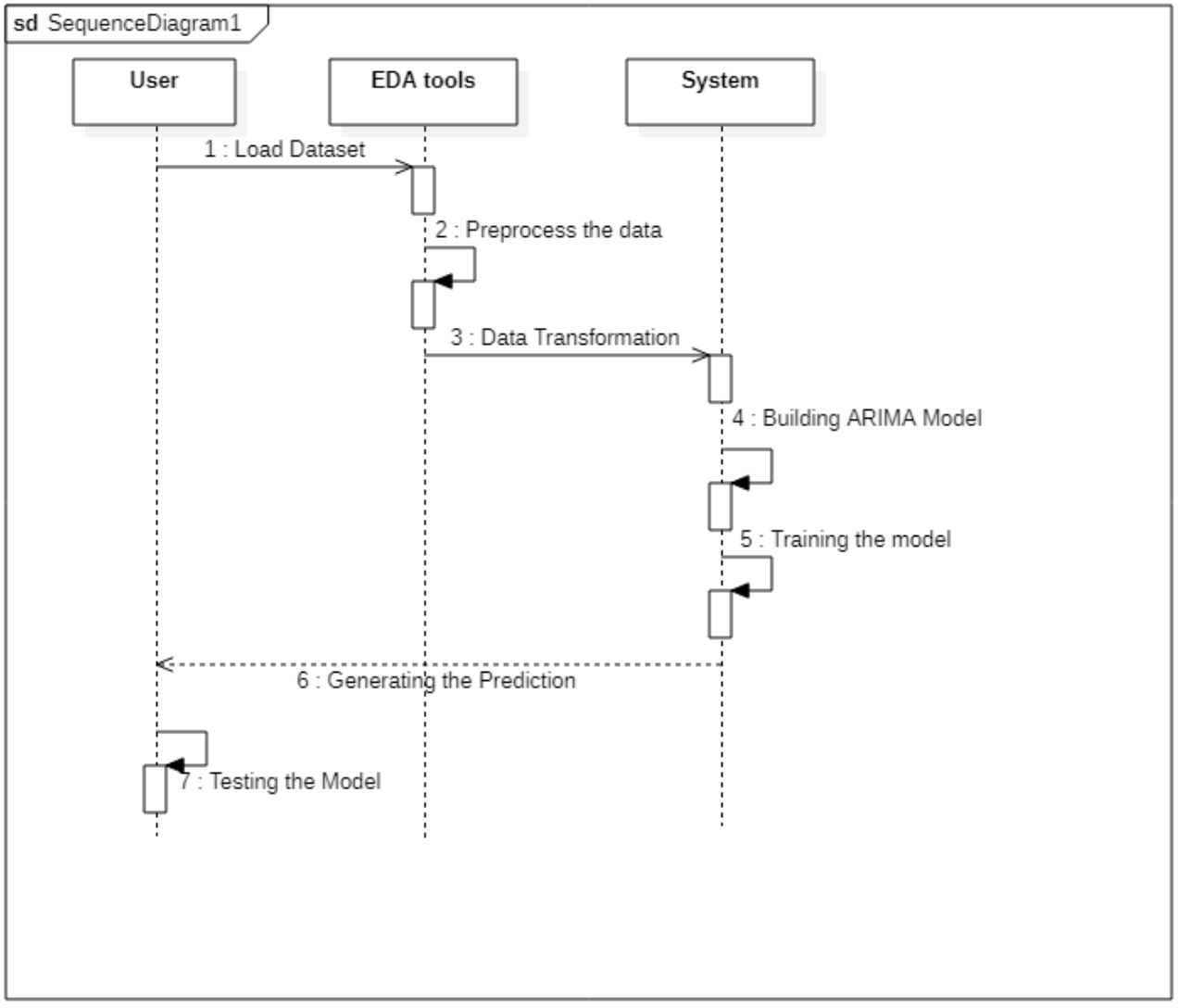


Fig. 3.7. Sequence of user interaction with the system through EDA tools

The Figure 3.7 shows the sequence followed in the project, where the user loads the dataset and preprocesses it. The system applies EDA tools for preprocessing, builds a ARIMA model, train the model and generate prediction. The results are generated back to the user interface and then we test the model.

3.3.4 Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

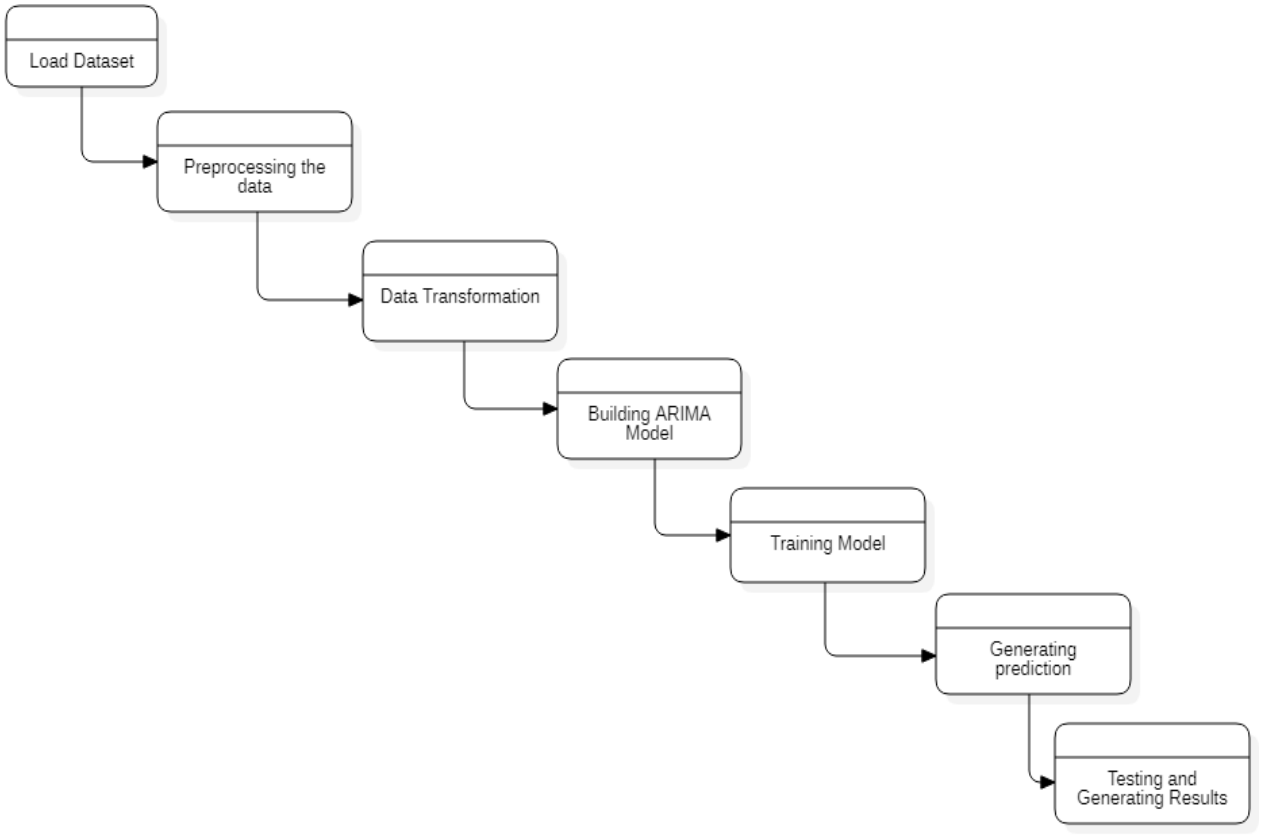


Fig. 3.8. Flow of Functionalities involved in project

The data flow diagram displayed in Figure 3.8 shows the flow of data right from loading dataset to data transformation, model building and generating the results.

3. TESTING AND RESULTS

4.1 Evaluation

4.1.1 Mean Squared Error (MSE)

For evaluating our model, we use Mean Squared Error (MSE). In deep learning, we have the concept of loss, which tells us how poorly the model is performing at that current instant. Now we need to use this loss to train our network such that it performs better. Essentially what we need to do is to take the loss and try to minimize it, because a lower loss means our model is going to perform better. MSE is calculated by taking the average of the square of the difference between the original and predicted values of the data.

Hence,

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(actual\ values - predicted\ values)^2 \ldots\ldots(4)$$

Here N is the total number of observations/rows in the dataset. The sigma symbol denotes that the difference between actual and predicted values taken on every i-value ranging from **1 to n**. This can be implemented using sklearn's mean_squared_error method.

Mean Squared Error is used to evaluate the regression problem's accuracy. As we are performing regression i.e., predicting the a and b channels to produce values of colored pixels, this evaluation metric is optimal. In statistics, the mean squared error (MSE) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss. The fact that MSE is almost always strictly positive (and not zero) is because of randomness or because the estimator does not account for information that could produce a more accurate estimate.

MSE is used to check how close estimates or forecasts are to actual values. Lower the MSE, the closer the is estimate to actual. This is used as a model evaluation measure for regression models and the lower value indicates a better fit.

4.1.2 Mean Absolute Error

Mean Absolute Error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. MAE is calculated as:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n} \quad \text{.....(5)}$$

It is thus an arithmetic average of the absolute errors |e|=|yi-xi|, where yi is prediction and xi is true value. Note that alternative formulations may include relative frequencies as weight factors. The mean absolute error uses the same scale as the data being measured. This is known as a scale-dependent accuracy measure and therefore cannot be used to make comparisons between series using different scales. The mean absolute error is a common measure of forecast error in time series analysis, sometimes used in confusion with the more standard definition of mean absolute deviation.

4.2 Results

Jupyter Notebook Environment

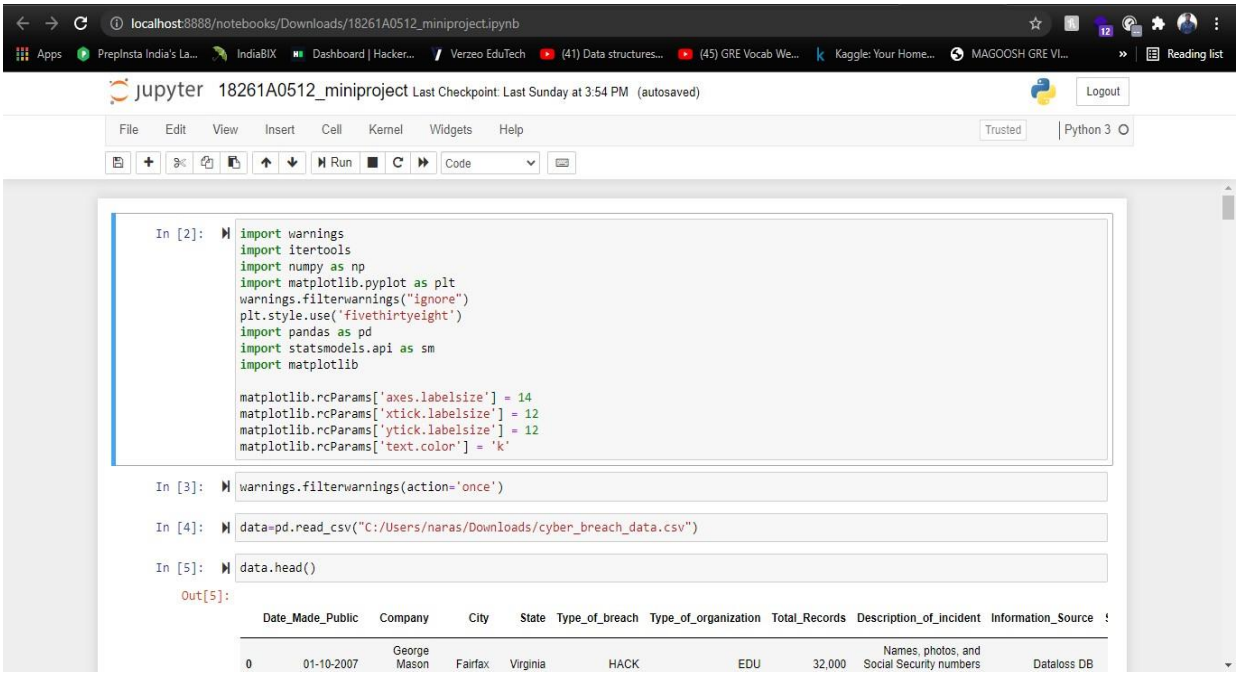


Fig.4.1. Jupyter Notebook Environment

The code for the “Prediction and Modelling of Cyber Hacking data breaches” is written here. The code is written in Python language.

As shown in the above figure 4.1, it shows the environment for the implementation of the working code.

4.2.1 Output

The total number of records has been split into various representations like trend, seasonal etc. It can be observed in fig.4.2.

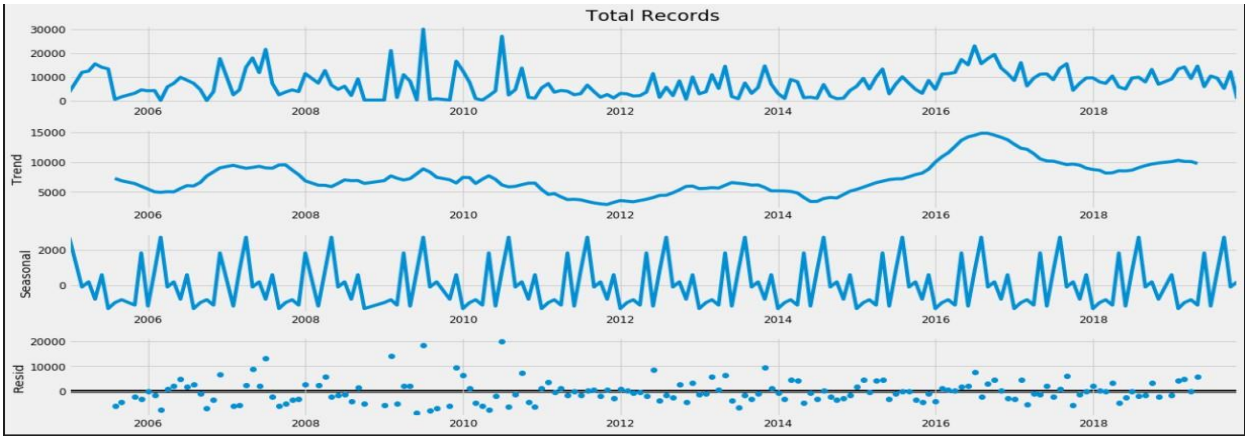


Fig.4.2. Total Records in various representations

```
results.summary()
```

SARIMAX Results

Dep. Variable:	Total_Records	No. Observations:	170
Model:	SARIMAX(0, 1, 1)x(0, 1, 1, 12)	Log Likelihood	-1457.249
Date:	Sun, 28 Nov 2021	AIC	2920.498
Time:	15:50:55	BIC	2929.387
Sample:	0	HQIC	2924.110
	- 170		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.8500	0.115	-7.411	0.000	-1.075	-0.625
ma.S.L12	-1.0052	0.126	-8.004	0.000	-1.251	-0.759
sigma2	7.306e+07	3.64e-10	2e+17	0.000	7.31e+07	7.31e+07

Ljung-Box (Q):	29.53	Jarque-Bera (JB):	79.69
Prob(Q):	0.89	Prob(JB):	0.00
Heteroskedasticity (H):	0.33	Skew:	1.19
Prob(H) (two-sided):	0.00	Kurtosis:	5.78

Fig.4.3. Summary of Model using SARIMAX Function

After training the model from 2014 to 2018, the prediction is generated for the number of total records breached from 2019 to 2021 using the SARIMAX function as shown in fig 4.3. The model is SARIMAX (0,1,1) x (0,1,1,12) and time stamp of the execution is shown. The AIC value is selected as minimum value from the set of values to better fitting of the model. The summary() function generates the total summary of the model which include model name, date,

time and correlations of variables with standard error and Statistical hypothesis values of variables at 0.025 and 0.975. The summary function tests the model using Ljung-Box test and Jarque-Bera test which generates the probabilities of errors in data.

```
In [30]: m=results.predict(start=pd.to_datetime("2019-01-01"),end=pd.to_datetime("2021-10-01"))
         m

Out[30]: Date_Made_Public
         2019-01-01    14034.632405
         2019-02-01    12923.098320
         2019-03-01    13527.514512
         2019-04-01    12480.268427
         2019-05-01    12349.142130
         2019-06-01    12055.154084
         2019-07-01    11822.525762
         2019-08-01    13285.129085
         2019-09-01    12704.583780
         2019-10-01    13089.624417
         2019-11-01    11533.690999
         2019-12-01    10707.603190
         2020-01-01    10127.325511
         2020-02-01     9837.742437
         2020-03-01     9526.376528
         2020-04-01     9312.033150
         2020-05-01     9848.334768
         2020-06-01     9057.242572
         2020-07-01     8256.413312
         2020-08-01    10434.900573
         2020-09-01     9649.999260
         2020-10-01     9158.304384
         2020-11-01     9675.571319
         2021-01-01     9065.693423
         2021-02-01     8660.230511
         2021-03-01     9172.117759
```

Fig.4.4. Predictions of the model.

In Figure 4.4, predictions of the model are generated as per the Date_Made_Public and these predictions are used for plotting the model.

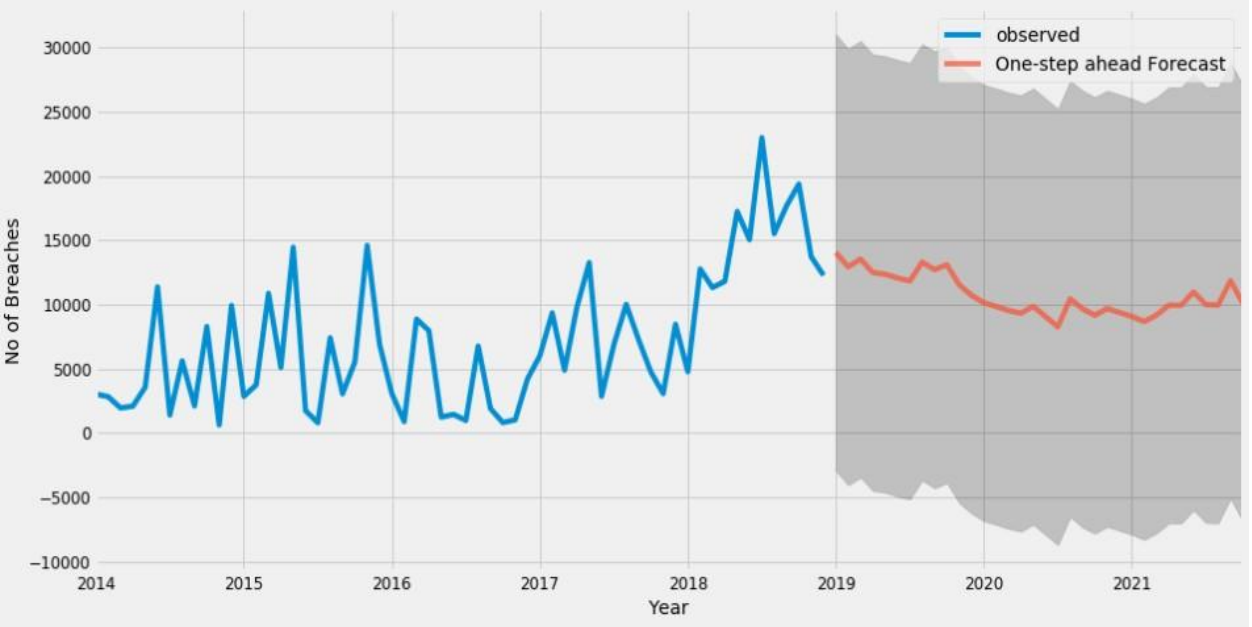


Fig.4.5. Modelling of Generated Prediction

In Figure 4.5, predictions of the model are plotted, the orange line represent one-step ahead Forecast that is our prediction and blue lines is the observed data.

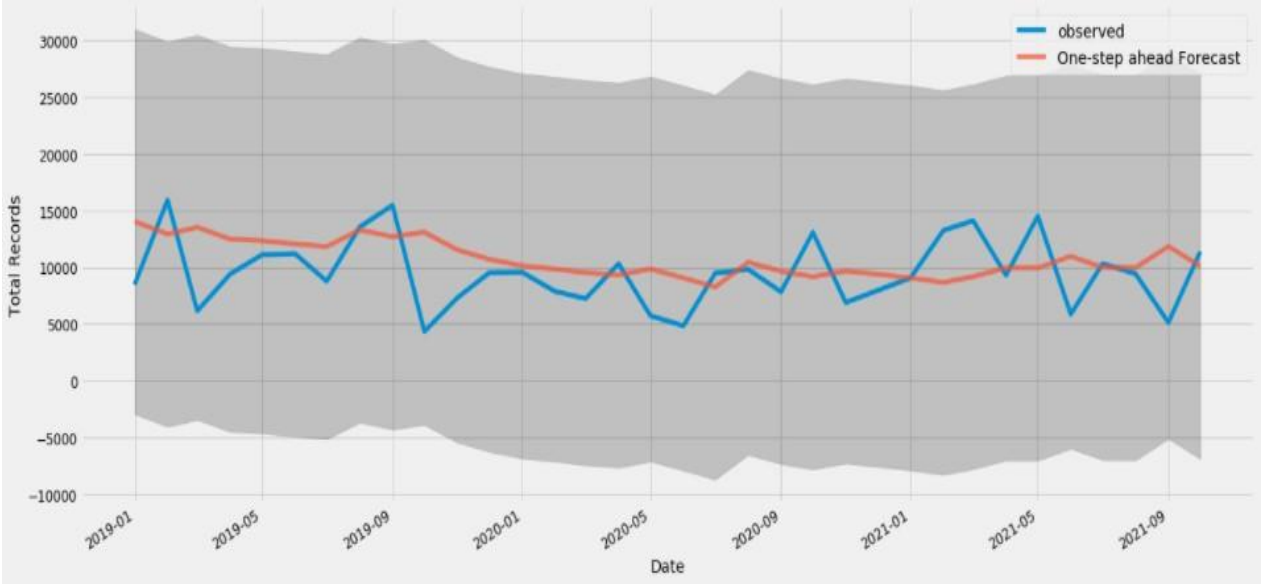


Fig.4.6. Total number of breaches for every 4 months

In Figure 4.6, predictions of the model are plotted for every four months, and the orange line represent predictions whereas blue represent observed data.

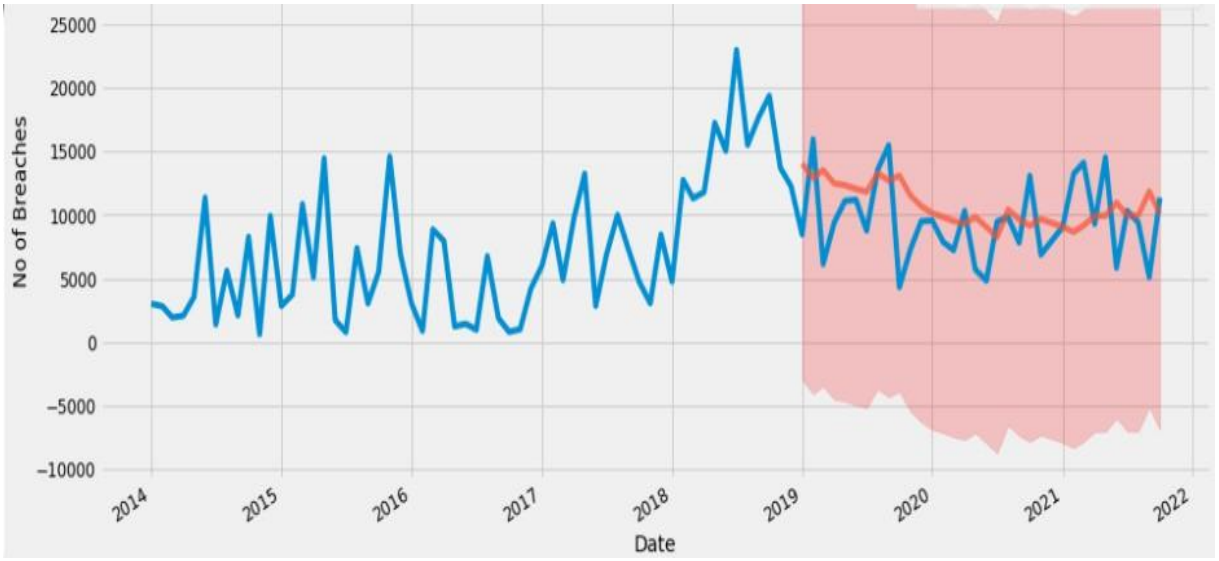


Fig.4.7. Result for number of breaches in year data

Figure 4.7 represents the prediction and observed data of breaches in a particular date and year.

4. CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

The proposed work analyzed a hacking breach dataset from the points of view of the incidents inter-arrival time and the breach size and showed that they both should be modelled by stochastic processes rather than distributions. The statistical models developed in this project show satisfactory fitting and prediction.

The prediction obtained from the Arima model can be used by the various institutions to get prepared for the future attacks on the organization. This model calculates the prediction based on moving average which results in accurate prediction. Testing the model by using various error metrics gives the error rate in prediction.

5.2 FUTURE SCOPE

Firstly, from the technical perspective, there is a need to assess new methods that threaten the security of critical information infrastructure. Secondly, from the perspective of law and policy, governments need to ensure that each sector identified as critical infrastructure should be properly protected both by legal and policy instruments. Further research is required to analyze the comprehensive legal landscape that aim to protect the critical information infrastructure, involving allenabling laws from all sectors.

BIBLIOGRAPHY

[1] R. R. Subramanian, R. Avula, P. S. Surya and B. Pranay, "Modeling and Predicting Cyber Hacking Breaches," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 288-293, IEEE, 2021.

[2] Z. Fang, M. Xu, S. Xu and T. Hu, "A Framework for Predicting Data Breach Risk: Leveraging Dependence to Cope With Sparsity," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2186-2201, IEEE, 2021.

[3] S. Fugkeaw, N. Kuasomboon and P. Panakitkul, "DBIM: An Efficient and Resilient PII Data Breach Incident Management System," 2021 13th International Conference on Knowledge and Smart Technology (KST), pp. 237-242, IEEE, 2021.

[4] D. Zegzhda, D. Lavrova and A. Khushkeev, "Detection of information security breaches in distributed control systems based on values prediction of multidimensional time series," 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), pp. 780-784, IEEE,2019.

[5] N. T. Cyriac and L. Sadath, "Is Cyber Security Enough- A study on Big Data Security Breaches in Financial Institutions," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), pp. 380-385, IEEE, 2019.

.

[6] Y. Fang, Y. Guo, C. Huang, and L. Liu, "Analyzing and Identifying Data Breaches in Underground Forums," in IEEE Access, vol. 7, pp. 48770-48777, IEEE, 2019

[7] Privacy Rights Clearinghouse (PRC) Dataset “https://www.kaggle.com/estratic/data-breaches-2007-2021”.

[8] ARIMA Model “https://ARIMA/blob/master/time-series-analysis-ARIMA”.

[9] First order autoregressive model “https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python”

[10] Arima random walk “https://people.duke.edu/411arim.html”

[11] SARIMAX Function “https://analyticsindiamag.com/sarimax-in-python-for-time-series-modeling”

[12] Mean Squared Error “https://en.wikipedia.org/wiki/Mean_squared_error”.

[13] Mean Absolute Error “https://en.wikipedia.org/wiki/Mean_absolute_error”.

APPENDIX

#In[1]

Download and unzip

https://www.kaggle.com/estratic/data-breaches-2005-2021

pip install tensorflow==1.15.0

pip install scikit-learn==0.22.1

pip install keras==2.2.4

pip install h5py==2.10.0

pip install pandas

pip install matplotlib

pip install opencv-contrib-python

pip install imutils

#In[2]

Download and import libraries

pip install tensorflow==1.15.0

pip install scikit-learn==0.22.1

pip install keras==2.2.4

pip install h5py==2.10.0

pip install pandas

pip install matplotlib

pip install opencv-contrib-python

pip install imutils

```
#In[3]
# Import Libraries
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
import pandas as pd
import statsmodels.api as sm
import matplotlib
matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 12
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'

#In[4]
# To display the first 5 rows in dataset
data=pd.read_csv("C:/Users/naras/Downloads/cyber_breach_data.csv")
data.head( )

#In[5]
#To find correlation between data
data.corr()
#In[6]
#To display the count of different types of breaches
data['Type_of_breach'].value_counts()
```

#In[7]

#Converting data into Time series data

```
time_series=data.loc[data['Type_of_breach'] == "HACK",
["Date_Made_Public","Total_Records","Type_of_breach"]]

time_series = time_series.dropna(subset=["Date_Made_Public"])

time_series["Total_Records"]=time_series["Total_Records"].str.replace(",","")

time_series["Total_Records"] = pd.to_numeric(time_series["Total_Records"], errors='coerce')

drop_indices=time_series["Total_Records"][time_series["Total_Records"]>30000].index

drop_index=time_series["Total_Records"][time_series["Total_Records"]==0].index

time_series=time_series.drop(drop_indices)

time_series=time_series.drop(drop_index)

time_series.index=range(1406)
```

#In[8]

#Grouping the Time series data based on Date_Made_Public

```
time_series = time_series.groupby('Date_Made_Public')['Total_Records'].sum().reset_index()

time_series = time_series.set_index('Date_Made_Public')

time_series.index = pd.to_datetime(time_series.index)

time_series.index
```

#In[9]

#Converting the data into Supervised data and plotting the graph

```
y = time_series['Total_Records'].resample('MS').mean()

y.plot(figsize=(15, 6))

plt.show()
```

```
#In[10]

#Decompostion plot of data

from pylab import rcParams

rcParams['figure.figsize'] = 18, 8

decomposition = sm.tsa.seasonal_decompose(y,freq=12, model='additive')

fig = decomposition.plot()

plt.show()


#In[11]
#Deciding p,d,q values range in ARIMA Model
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]
print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))


#In[12]
 #Train and Test data
 train_data = y['2014':'2018']
 test_data = y['2019':'2021']

#In[13]
#Training the data with SARIMAX Function
```

```
AIC=[]

SARIMAX_model = []

for param in pdq:

for param_seasonal in seasonal_pdq:

try:

mod =
sm.tsa.statespace.SARIMAX(y,order=param,seasonal_order=param_seasonal,enforce_stationarity=
False,enforce_invertibility=False)

results = mod.fit()

print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))

#print('SARIMAX{}x{} - AIC:{}'.format(param, param_seasonal, results.aic), end='\r')

AIC.append(results.aic)

SARIMAX_model.append([param, param_seasonal])

except:

continue


#In[14]

#Selecting the Minimum AIC for the best fitting of model.

print('The smallest AIC is {} for model SARIMAX{}x{}'.format(min(AIC),
SARIMAX_model[AIC.index(min(AIC))][0],SARIMAX_model[AIC.index(min(AIC))][1]))



#In[15]

#Fitting the model

mod =
sm.tsa.statespace.SARIMAX(y,order=SARIMAX_model[AIC.index(min(AIC))][0],seasonal_order
=SARIMAX_model[AIC.index(min(AIC))][1],enforce_stationarity=False,enforce_invertibility=Fa
lse)

results = mod.fit()
```

```
#In[16]
#Printing summary of the model
results.summary()

#In[17]
#plot diagnostic plots for knowing whether the data contain any outliers
results.plot_diagnostics(figsize=(20, 14))
plt.show()

#In[18]
#Generating prediction from 01-01-2019 to 01-10-2021
pred = results.get_prediction(start=pd.to_datetime('2019-01-01'), dynamic=False)
pred_ci = pred.conf_int()

#In[19]
#Breach prediction results are printed
m=results.predict(start=pd.to_datetime("2019-01-01"),end=pd.to_datetime("2021-10-01"))
m

#In[20]
#plotting the prediction
pred = results.get_prediction(start=pd.to_datetime('2019-01-01'), dynamic=False)
pred_ci = pred.conf_int()
ax = train_data.plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7))
ax.fill_between(pred_ci.index,pred_ci.iloc[:, 0],pred_ci.iloc[:, 1], color='k', alpha=.2)
```

```
ax.set_xlabel('Year')
ax.set_ylabel('No of Breaches')
plt.legend()
plt.show()

#In[21]
#plot for breach validation
ax = y['2019':].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7)
ax.fill_between(pred_ci.index,pred_ci.iloc[:, 0],pred_ci.iloc[:, 1], color='k', alpha=.2)
ax.set_xlabel('Date')
ax.set_ylabel('Total Records')
plt.legend()
plt.show()

#In[22]
#plot of record breaches in year wise
pred = results.get_prediction(start=pd.to_datetime('2019-01-01'), dynamic=False)
pred_ci = pred.conf_int()
ax = y['2014':'2022'].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7))
ax.fill_between(pred_ci.index,pred_ci.iloc[:, 0],pred_ci.iloc[:, 1], color='red', alpha=.2)
ax.set_xlabel('Date')
ax.set_ylabel('No of Breaches')
plt.legend()
plt.show()
```

```
#In[23]

#Finding Mean Squared Error of plot

y_forecasted = pred.predicted_mean

y_truth = y['2019-01-01':]

mse = ((y_forecasted - y_truth) ** 2).mean()

print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2)))


#In[24]

#Finding Mean Error

mean_error=(y_forecasted-y_truth).sum()/len(y_truth)

print('Mean Error:',mean_error)
```