Network Modeling Analysis in Health Informatics and Bioinformatics (2022) 11:52
https://doi.org/10.1007/s13721-022-00391-1

ORIGINAL ARTICLE

Using attack graphs to defend healthcare systems from cyberattacks: a longitudinal empirical study

Hüseyin Ünözkan[1] · Mehmet Ertem[2] · Salaheddine Bendak[3]

Received: 6 August 2022 / Revised: 1 October 2022 / Accepted: 19 October 2022 / Published online: 16 November 2022

Abstract

Cyber security encompasses a variety of financial, political, and social aspects with significant implications for the safety of individuals and organisations. Hospitals are among the least secure and most vulnerable organisations in terms of cybersecurity. Protecting medical records from cyberattacks is critical for protecting personal and financial records of those involved in medical institutions. Attack graphs, like in other systems, can be used to protect medical and hospital records from cyberattacks. In the current study, a total of 352 real-life cyberattacks on healthcare institutions using common vulnerability scoring system (CVSS) data were statistically examined to determine important trends and specifications in regard to those attacks. Following that, several machine learning techniques and an artificial neural network model were used to model industrial control systems (ICS) vulnerability data of those attacks. The average vulnerability score for attacks on healthcare IT systems was found to be very high. Moreover, this score was found to be higher in healthcare institutions which have experienced cyberattacks in the past and no mitigation actions were implemented. Using Python programming software, the most successful model that can be used in modelling cyberattacks on IT systems of healthcare institutions was found to be the *K*-nearest neighbours (KNN) algorithm. The model was then enhanced further and then it was tried to make predictions for future cyberattacks on IT systems of healthcare institutions. Results indicate that the overall score is critical indicating that medical records are, in general, at high risk and that there is a high risk of cyberattacks on medical records in healthcare institutions. It is recommended, therefore, that those institutions should take urgent precautionary measures to mitigate such a high risk of cyberattacks and to make them more secure, reliable, and robust.

Keywords Internet of medical things · Healthcare systems · Industrial control systems · Machine learning · Cyber-attacks

1 Introduction

Although transactions and communications over the internet have been occurring for more than quarter of a century, the COVID-19 pandemic has helped in accelerating them (Majeed and Lee, 2021; Pollini et al. 2022). This acceleration in digital transformation has been recorded in almost all sectors and has resulted in significant changes in working habits, commercial dealings, and supply chain activities (Desruelle et al. 2019; Pollini et al. 2022; Teal 2020).

Cyber security incorporates several financial, political and social aspects and has tremendous implications for the safety of individuals and organisations (Lallie et al. 2020). In cyberattacks, the attacker aims to obtain usually confidential information on one or more target hosts including computers, routers, firewalls, databases and other network components. Such attacks might incorporate multiple hosts and can be used as crossing means for further larger attacks with the aim to reach a target host. Graph theory can be used to represent cyberspace as the basic graph structure imitates the interconnectivity of computer networks (Angel 2022; Lallie et al. 2020).

✉ Salaheddine Bendak
sbendak@halic.edu.tr

Hüseyin Ünözkan
huseyinunozkan@halic.edu.tr

Mehmet Ertem
mertem@ogu.edu.tr

[1] Department of Industrial Engineering, Haliç University, Eyüpsultan, Istanbul, Turkey

[2] Department of Industrial Engineering, Eskişehir Osmangazi University, Eskişehir, Turkey

[3] Department of Industrial Engineering, Haliç University, Eyüpsultan, Istanbul, Turkey

One of the weakly secured and vulnerable organisations in terms of cybersecurity are hospitals. Hospitals have a full load of personal information like detailed health records, patient and staff addresses, credit card details, identification numbers, social security records, death records, etc. The COVID-19 pandemic has even accelerated this collection of health and private data to combat its spread. Information sought and recorded during the pandemic included whom we see, where we go, what we eat, whom we work with, etc. In parallel, digital developments over the last few years [reflected through internet of things (IoT), internet of medical things (IoMT), cloud computing, mobile phone health-related apps, etc.] increased people's reliance on digital storage of private data (Majeed and Lee 2021). In the meantime, the ever-increasing connectivity of medical devices to external systems over the internet increases the risk of cyberattacks (Islam et al. 2022).

Such private data are highly sought after data by attackers and hackers. At the same time, health information is multidimensional (with a multitude of attributes related to the human body) and highly dynamic in nature (keep changing over time in most cases). These attributes make defending healthcare systems even harder. Unfortunately, most hospitals do not have adequate resources to monitor and defend threats to their systems. Many of them lack the manpower to maintain a full cybersecurity system, awareness of the importance of those systems and cybersecurity programs that can detect network activity and point out any intrusion attempts (Angel 2022; Coventry and Branley 2018; Majeed and Lee 2021; Teal 2020).

Attack graphs represent a popular cyberattack mitigation modelling technique that is used in defending online systems. They are illustrations that are used to determine if designated endpoints, like servers, can be reached by attackers trying to infiltrate computer networks. In general, they are graphs where the starting node denotes an attacker, and nodes and arcs represent actions the attacker tries to take and alterations in the network state triggered by these actions. Those actions usually incorporate steps that exploit vulnerabilities in the network. Attack graphs are used to capture and illustrate those paths that can be used by attackers in a visual way and to minimise the cognitive load on cybersecurity experts trying to develop methods to defend the systems (Durkota et al. 2019; Ertem and Bier 2021; Kaynar 2016; Lallie et al. 2020; Sheyner et al. 2002).

A complete attack graph shows all possible sequences of attacker actions that potentially can lead to obtaining secured information on the target side. The literature on attack graphs show that some authors use nodes to represent network states and arcs to represent attack actions, while others use other representations. Some attack graphs have one attacker starting location and one target host, while some others have multiple starting locations for attackers and/or multiple targets (Kaynar 2016; Lallie et al. 2020; Sheyner et al. 2002).

Defending hospital records from being leaked is vital for protecting personal and financial records of individuals involved in those hospitals. Like the case in other systems, attack graphs can be incorporated in defending hospital records from cyberattacks. This issue has not been fully investigated in the peer-reviewed literature, as was also postulated by Lallie et al. (2020).

This study aims to assess real-life cyberattacks on health records using data mining techniques, analyse their attributes and develop ways on how attack graphs can be used in combating those attacks. It is believed that shedding light on the use of attack graphs in defending healthcare systems will help in developing cyber defence mechanisms or improving existing ones in those systems. In the next section, materials and methods are introduced. Following that results and discussion are given followed by the conclusions section.

2 Material and methods

In this study, 352 cyberattacks on components within the scope of Industry 4.0 used in the healthcare sector, were used. The attacks are in vector form and are based on real-life observations. The data includes vulnerabilities of industrial control systems (ICS) from common vulnerability scoring system (CVSS) in the United States. The dataset contains information from CVSS v3 base score calculations obtained from Cybersecurity and Infrastructure Security Agency (CISA). The dataset included observations between January 1999 and May 2022.

Descriptive statistical analysis results show clearly that companies which have high and critical level scores have experienced cyberattacks in the past and no mitigation actions were implemented. It was also found that healthcare and medical institutions that took action to increase the reliability of their systems after cyberattacks were significantly less vulnerable to future cyber penetrations that those who did not take any action after previous attacks.

ICS data, that use metrics of the common vulnerability scoring system (CVSS), is a unique subset of vulnerability data within the CVSS database (National Institute of Standards and Technology 2022). ICS has been shown to be more vulnerable to cyberattacks and constitutes a different group than the general CVSS for all computer systems. CVSS consists of three metric groups: baseline, temporal and environmental. Baseline metrics represent the intrinsic properties of vulnerability which are "fixed" over time and across user environments. The temporal set of metrics reflects the properties of vulnerability that may change over time but may not change in user environment. Environmental set of metric group indicates how the seriousness of vulnerability changes by virtue of altering certain aspects of an organisation.

To maintain the desired level of effectiveness, organisations seek ways to improve their security posture by identifying potential vulnerabilities in their systems to prioritise investment in cybersecurity. To achieve this, organisations commonly use the common vulnerability scoring system (CVSS) to evaluate the importance of potential vulnerabilities. CVSS ensures a standard measurement and catches key results of software and hardware vulnerabilities.

Mitigation for IT vulnerabilities commonly includes coding changes but can also involve feature changes or even feature resistance (National Institute of Standards and Technology 2022). In this context, each Industry 4.0 component including critical systems, industrial control systems (ICS), cell phones and home-usage devices like smart cleaners are vulnerably exposed to these threats.

This study aims to use advanced optimisation techniques to model ICS vulnerability data based on CVSS. To achieve this goal, some commonly used machine learning (ML) algorithms and an artificial neural network model were used. The most successful model is determined as the proposed model and is then improved by incorporating some additional coding to improve the model in Python.

CVSS has a pre-determined formula used to evaluate risk scores on CVSS data (first.org 2022). Some studies in the literature tried to assess CVSS and offer various models. For instance, Dondo (2008) recommended a fuzzy system, Anikin (2017) suggested a risk assessment with fuzzy method, Lorenzo et al. (2020) developed a risk assessment model using CVSS data, Wang et al. (2020) proposed a Bayesian attack graph for CVSS which offered paths to determine attacker's ability to forecast the success probabilities of attacks using CVSS data. Moreover, Keramati and Akbari (2013) suggested a different graph model for CVSS metrics and Zhang et al. (2017) used conditional probability to determine effectiveness of vulnerabilities on CVSS. Also, Wu et al. (2019) suggested a principal component analysis model for improving CVSS and Khazaei et al. (2016) compared support vector machines, random forest algorithms and fuzzy methods for CVSS. Finally, Ertem and Bier (2021) generated a stochastic model with game theory specialties to reach a conclusion about attackers' paths in cyberattacks. They defined some rules in their stochastic model and under these assumptions they successfully estimated attackers' behaviours.

In this study, SPSS and Python programs were used to develop a model to estimate vulnerability of cyber systems in healthcare institutions. Using SPSS, ordered logistic regression, and using Python, linear regression, ordered logistic regression, decision tree, random forest, *K*-nearest neighbours (KNN) and artificial neural network algorithm Keras deep learning were proposed for evaluating and assessing such vulnerabilities. In the next subsections, those algorithms are explained. Since KNN is found to be the most promising model as explained later, it is presented in more details.

2.1 CVSS scoring system

CVSS scoring system consists of five basic components (with categorical variables consisting of eight basic components) to calculate a score: attack vector (AV), attack complexity (AC), privileges required (PR), user interaction (UI), scope, confidentiality (C), integrity (I), availability (A) (see Table 1).

Attack vector: This variable can take value as, network, adjacent, local, and physical. Attack vector shows the

Table 1 Sample of CVSS scores

Observation	Category	AV	AC	PR	UI	Scope	C	I	A	Score
1	High	Local	Low	None	Required	Unchanged	High	High	High	7.8
2	High	Network	Low	None	None	Unchanged	None	None	High	7.5
3	Critical	Network	High	None	None	Changed	High	High	High	9
4	High	Local	Low	Low	None	Changed	High	High	High	8.8
5	High	Network	Low	None	None	Unchanged	High	None	None	7.5
6	High	Network	Low	Low	None	Unchanged	High	High	High	8.8
7	Medium	Network	High	None	Required	Changed	None	High	None	6.1
8	High	Local	High	Low	None	Changed	None	High	High	7.5
9	Critical	Network	High	None	None	Changed	High	High	High	9
10	Medium	Network	Low	High	Required	Unchanged	High	High	High	6.8

information of exploitation about possible vulnerability. Network represents remote exploit over internet. Physical represents the attacker assault a component via physically touch.

Attack complexity: This variable can take value as low or high. Attack complexity shows the situations about attacker's control which has to exist to reach vulnerability. It takes value as "High" when the attacker invests some measurable effort in preparation for vulnerable component before an attack. When the attack complexity is low, the score is high, because attack can easily be executed.

Privileges required: This variable can take value as none, low, high. Privileges required show privilege level which the attacker has to possess before a successful attack. If no privilege is required, the score is high, because anybody can exploit the system.

User interaction: This variable can take value as none or required. User interaction shows requirement of any user interaction. If user interaction is not required, the score will be higher.

Scope: This variable can take value as changed or unchanged. Scope shows whether vulnerability is related to security scope. Generally, in a regular organisation, each component is under a determined security authority, means under the same single jurisdiction of a security scope.

Confidentiality: This variable can take value as high, low or none. Confidentiality measures the effect of the reliability of information sources, led by a software device due to a successful attack. It will be high, when there is an unreliability (e.g. attacker gains user password).

Integrity: This variable can take value as high, low, none. Integrity calculates the effect of integrity in a successful attack. Integrity takes value as high once the attacker can modify any files protected by the affected device.

Availability: This component can take value as high, low, none. It calculates availability of the affected component after attack. It will be high once the attacker can deny some availability. For instance, if the attacker filled all the memory usage area, the system is still available but cannot be used (first.org 2022; Gencer and Başçiftçi 2020).

Once the base metrics are assigned values by an analyst in CVSS calculation, the base score will range from 0.0 to 10.0 and this rating is later convert into categories (none, low, medium, high, and critical). In this study, 352 observations from medical systems of Industry 4.0 component attack vectors of CVSS were used. The main target was to reach a reliable model for predicting possible cyber assaults on healthcare systems in the future so that mitigations and precautions can be developed and implemented more successfully.

2.2 Utilised algorithms

KNN algorithm is one of the most popular algorithms in data mining. KNN is an easy but robust classification technique. It does not require training to make prediction, which is generally the hardest stage of ML. KNN has been widely used to determine similarity and path recognition. In addition, it has also been used to develop recommendations for dimension reductions and pre-steps in virtual vision, especially for face matching tasks. KNN, which gained more popularity after taking part in machine learning algorithms, tries to find distances from determined query to examples in datasets, choosing the decided number of observations nearest to this query, in terms of this work, measure frequency and determine label metrics in query. Because of this, KNN algorithm contains a lazy piece, determining K value and finding K nearest neighbours. Thus, KNN classification has a different learning process, which does not require training dataset (Zhang 2011).

Cover and Hart (1967) proposed the KNN classification algorithm. After this, many research studies have utilised it and and some improvements on the initial algorithm have been suggested (Chen et al. 2020; Zheng et al. 2020). When a model is generated with KNN algorithm, K value determination and nearest neighbor query are two crucial issues. The nearest neighbor can be determined by different distance measurement functions, like Manhattan distance, Mahalanobis distance, Euclidean distance and angle cosine distance. K value determination can be determined through cross-validation methods or expert settings. K value determination does not directly affect the problem solving but it affects success rate of determination via changing learning methodology. When K value is chosen to be very small, it can cause overfitting. Selecting a very large K value can increase the error of learning (Durbin et al. 2020). Because of this, researchers have offered different techniques to gain optimal K value calculation (Li et al. 2003).

To determine nearest neighbour, researchers have offered different measurement techniques for the distance measurement function. For example, Abu-Aisheh et al. (2020) suggested a new distance measurement for KNN after which this algorithm became even more powerful with shorter manipulation durations when the number of variables is

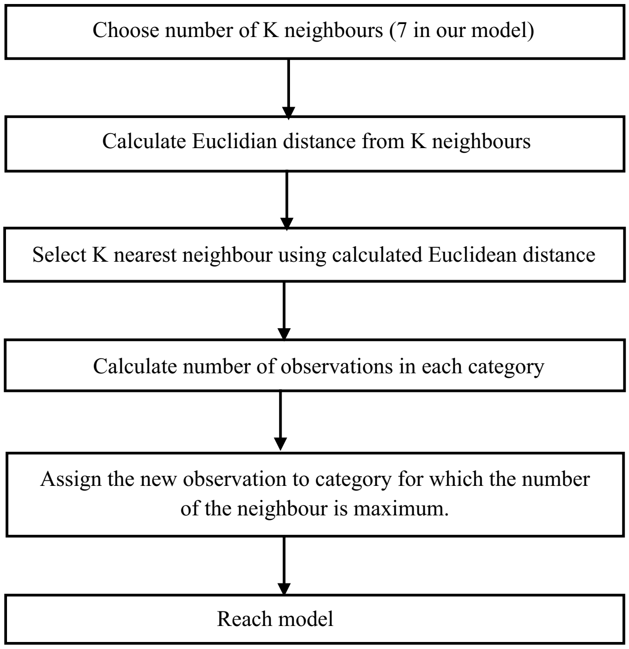

Choose number of K neighbours (7 in our model)
Calculate Euclidian distance from K neighbours
Select K nearest neighbour using calculated Euclidean distance
Calculate number of observations in each category
Assign the new observation to category for which the number of the neighbour is maximum.
Reach model

Fig. 1 KNN flowchart

Table 2 Scores of companies with critical and high-level CVSS

Companies	Av. score	Observation no.	Mitigation no.	Non-mitigated attacks
Company A	9.8	1	1	Yes
Company B	9.8	1	1	Yes
Company C	9.8	6	6	Yes
Company D	9.6	33	33	Yes
Company E	9.4	8	8	Yes
Company F	9.2	2	2	Yes
Company G	8.9	1	1	No
Company H	8.65	2	2	Yes
Company I	8.63	8	8	Yes
Company J	8.2	14	14	No
Company K	8.0	11	11	No
Company L	7.8	5	5	Yes
Company M	7.6	1	1	No
Company N	7.3	1	1	Yes
Company O	7.3	1	1	No
Company P	7.2	17	17	No
Company R	7.0	8	8	No

large. Gou et al. (2019) suggested a KNN algorithm via calculation with means of local situations. In their algorithm, a new function for distance measurement is generated with a linear combination between test and train data sets.

In Python, KNN algorithm analysis has some variables such as metric variable which can take values such as Euclidean, Manhattan, Mahalanobis and cosine. There are default euclidean values for this variable which were used in the current study to reach sufficiently successfull model with KNN (Chomboon et al. 2015).

The Euclidean measurement for distance to reach a value between two points is defined as

$$d^2st = \left(x_s - y_t\right)\left(x_s - y_t\right)'. \tag{1}$$

A simple KNN flowchart for Euclidean measurement procedure followed in the current study is given in Fig. 1.

There are three different logistic regression methods: binary, ordinal and nominal logistic regression. In linear regression, the explained variable is continuous; appropriate model is trying to predict explained variable's value, and explanatory variable must be normally distributed. If errors of estimation are normally distributed with a constant variance, it can be assumed that there will be a significant linear regression model (Yilmaz and Unozkan 2015).

ML pioneers defined it as a "field of study that gives computers the ability to learn without being explicitly programmed." Thus, this analysis centers on classification and estimation and on ground properties which were known previously via train dataset (Buczak and Guven 2016). ML analysis commonly consists of two phases: training and test phases. Generally, steps performed are as follows (Buczak and Guven 2016):

1. Define classes from train dataset.
2. Define a subset of features which are necessary.
3. Teach machine using train dataset.
4. Try to gain success rate of this trained model with test dataset.

Under the criteria of partition or stopping, decision trees may be used for both classification and regression. With an optimisation definition, variables split to internal nodes according to determined criteria. The most wide-spread criteria for classification is entropy which depends on lower bound on the length of a random variable's bit representation (Kaun et al. 2021).

Random forest is a high-performing advanced optimisation method in learning algorithms. Especially for social sciences, this algorithm can provide wide usage area with extend modelling capability. The random forest algorithm performs better in estimation of error rates than decision

Table 3 Model success rates in medical CVSS dataset

Program	Model	Success (%)
SPSS	Ordinal logistic regression	59.2
Anaconda	Ordinal logistic regression	59.7
Anaconda	Lineer regression	72.6
Anaconda	Artificial neural network	84.3
Anaconda	Decision tree	85.3
Anaconda	Random forest	86.0
Anaconda	KNeighbors	87.3

trees. The error of analysis in random forest is calculated via the out-of-bag error while the training process is carried out (Schonlau and Zou 2020).

3 Results and discussion

As explained earlier, the current study incorporated statistically examining CVSS data in the United States and suggested mitigation actions. Then a model was developed in the current study with the aim of giving a specific and useful tool to decision makers for evaluating previous attacks and predicting future cyberattacks on healthcare institutions.

3.1 Phase 1: descriptive statistics

In Table 2, information on companies which had high and critical level score averages is given. According to Table 2, it was observed that all companies with critical level average score points have at least 1 non-mitigated attack. In addition 4 out of 11 companies with high average score have had at least 1 non-mitigated attack. This shows that non-mitigated attacks serve as a strong indictor to higher CVSS score and IT vulnerability. When such scores start to increase, reliability of Industry 4.0 systems and CVSS scores decrease in general.

These results clearly indicate that mitigation of cyberattacks and CVSS scores are adversely correlated. This means that when a company takes precautionary measures after any cyberattack on healthcare institutions in the United States, cyberattack vulnerability score tends to decrease.

As clearly can be seen in Table 2, all variables which are used in CVSS calculation are categorical variables. Thus, each variable in the developed model should have its 'own significance.' In this study, each investigated model involved the same type of variables, i.e. categorical random variables. Because the structure of these variables is categorical, it is believed classification models, cluster analysing models or novel optimisation techniques like ML algorithm or artificial neural network algorithm with Keras are better options than classical statistical models.

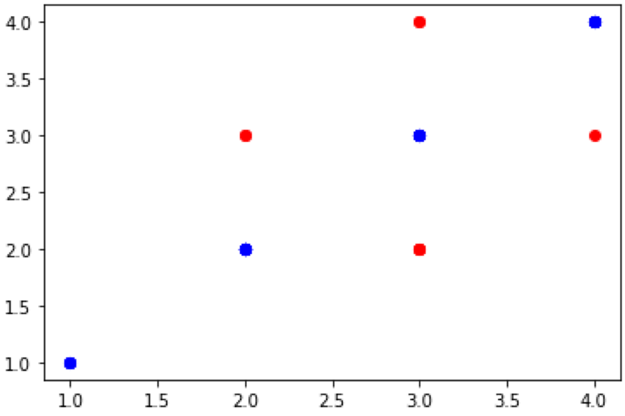


Fig. 2 KNN prediction graph

3.2 Phase 2: statistical analysis

While analysing the CVSS dataset, 80% of the dataset was used for training and 20% as test data in all models. Model success rates are given in Table 3.

Out of the seven models developed for assessing and predicting attacks on healthcare IT systems in Table 3, it was found that the most successful model in predicting cyberattacks was the KNN algorithm with success rate by 87%. Therefore, it was decided to focus on this model and try to develop it further.

Python codes for each model in Table 3, are given in Appendix 1. For developing the KNN algorithm further, additional coding for basic categorical cluster needs like stratified train and for a SoftMax activation function in the last layer and 'sparse_categorical_cross-entropy' function for loss evaluation was needed.

This high success rate of KNN in modelling CVSS datasets was achieved with seven neighbours. The model can also help in determining probable effect of any future cyberattacks on healthcare systems. Thus, an alternative score definition model for CVSS was developed. In Fig. 2,

Table 4 Test dataset and predictions

Category	Test dataset	Correct prediction
Low	4	4
Medium	34	32
High	18	21
Critical	15	14

a KNN prediction graph for the developed model is given. In Fig. 2, blue dots represent successful predictions and red dots represent unsuccessful predictions. In this analysis, 71 observations (attacks) were used as a test dataset. The success rate recorded is 87%. Test data and predictions are given in Table 4.

Over 71 observations in test dataset, 62 observations were predicted correctly. The 9 failure predictions were distributed to medium, high and critical level observations. This shows that the proposed model to predict future cyberattacks is reliable and does not exhibit any bias in predictions.

Based on predictions done using the developed model, the most frequently seen observations in each variable are given in Table 5.

When estimating a score value with these most observed values, the score category was found to be critical. Therefore, it was concluded that there is potentially a high risk of cyberattacks on healthcare institutions and that those institutions should pay more attention to cyberatack risk mitigation processes.

4 Conclusion

In this study, 352 real-life cyberattacks on healthcare systems of Industry 4.0 components of CVSS were used. Analysis of this dataset was done in two phases as postulated earlier. In the first phase, it was found that healthcare institutions that took action to increase the reliability of their systems after cyberattacks were significantly less vulnerable to future cyber penetrations that those who did not take any action after previous attacks. In the second phase, and to reach a successful prediction model, several machine learning techniques and an artificial neural network model were utilised to assess industrial control systems vulnerability data based on common vulnerability scoring system scores. The most successful model was determined and was developed further by utilising some additional codes for model improving using Python.

Out of the several models developed for assessing and predicting attacks on healthcare IT systems, it was found that the most successful prediction model was the KNN algorithm with success rate of 87%. The developed model included a stratified training dataset, a SoftMax activation function in the last layer and a 'sparse_categorical_cross-entropy' function for loss evaluation with seven neighbours.

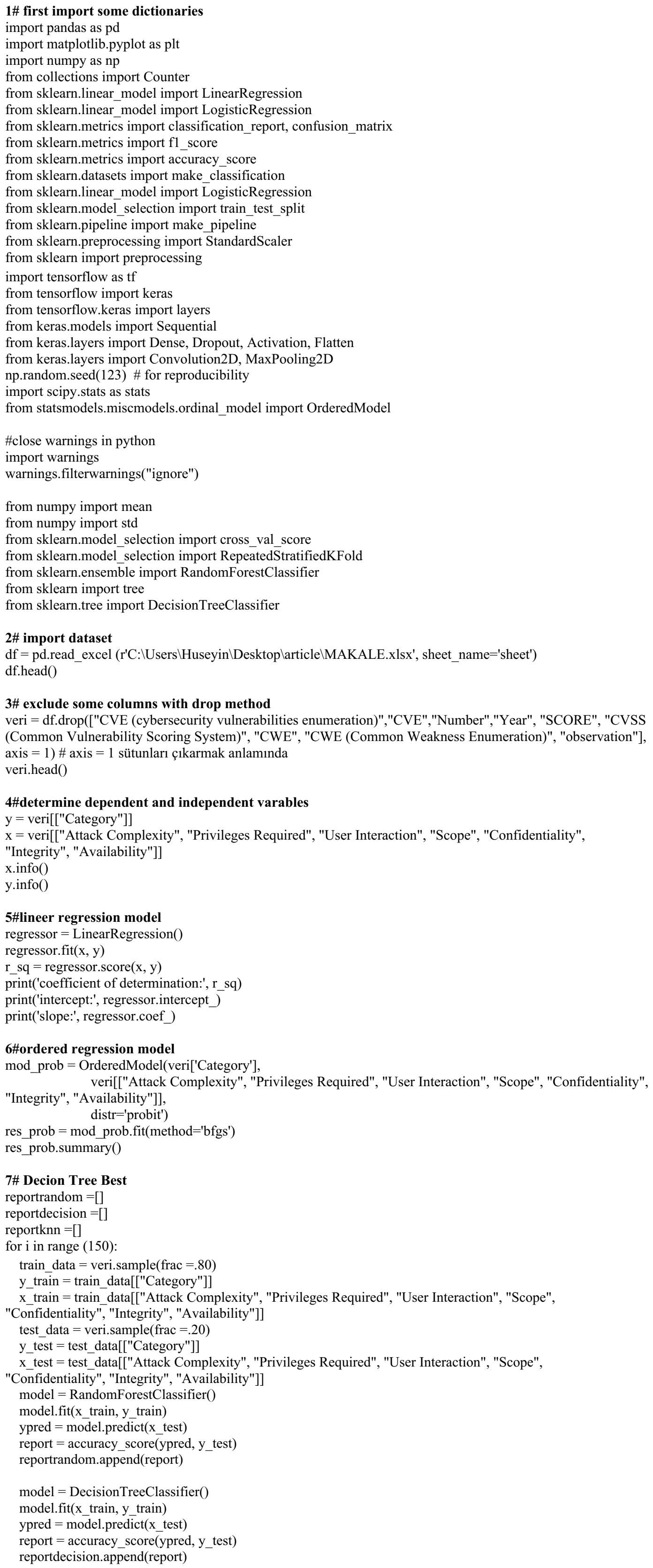
Based on the results of the current study, it can be clearly concluded that the average score of attacks on healthcare IT systems is, in general, very high. This score is even higher in institutions which experienced cyberattacks in the past and did not implement any mitigation actions. This situation involves a very high risk on human lives, especially for critical healthcare IT systems like oxygen supplying equipment and hospital baby incubators. In general, it can be concluded that there is a high risk of cyberattacks on healthcare institutions and those institutions should take precautionary measures to minimise the risk of cyberattacks and defend their systems.

As a limitation of the current study, it should be noted that the open-source cyberattacks dataset used included attacks only in the United States. Also, the dataset was somehow a small one with 352 attacks. It is recommended that future studies try to include bigger datasets and from different countries.

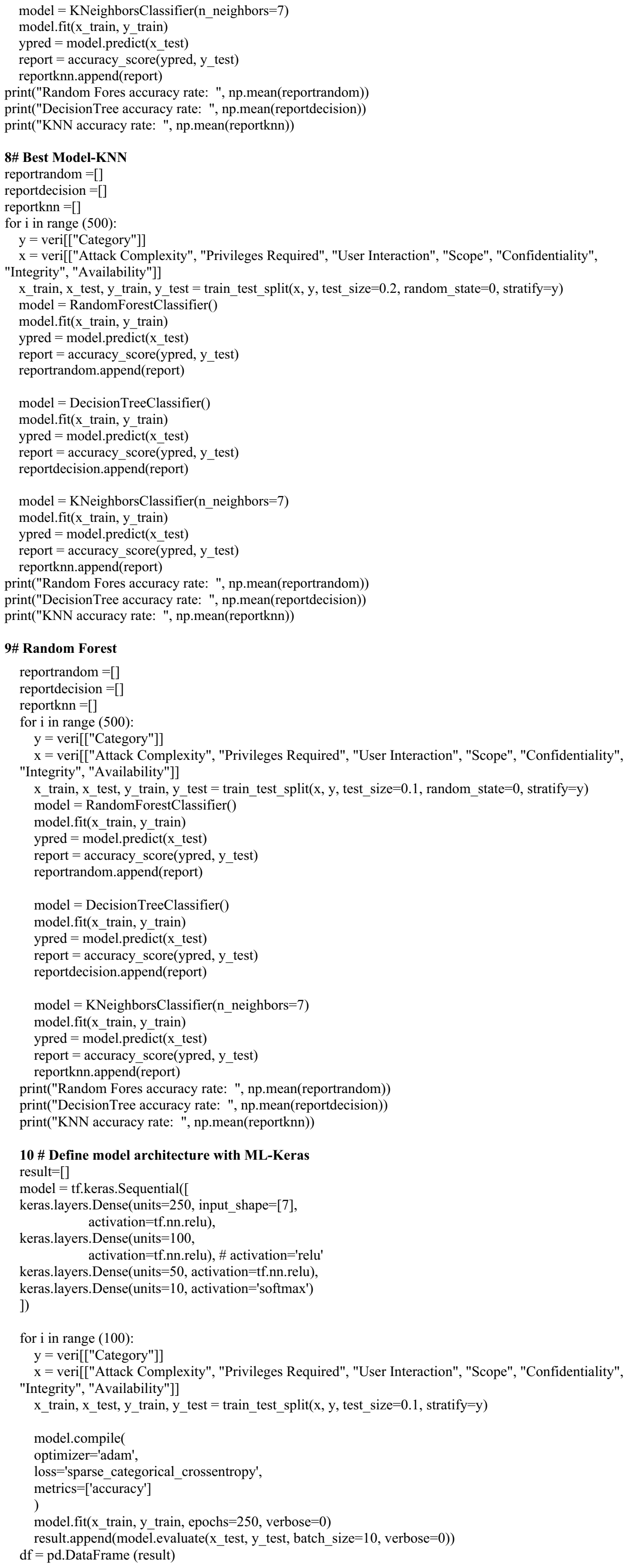
Table 5 Most seen observations of cyberattacks on healthcare institutions

AC	PR	UI	Scope	C	I	A
Low	None	None	Unchanged	High	High	High

Appendix 1: PYTHON codes



```
1# first import some dictionaries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from collections import Counter
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn import preprocessing
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D, MaxPooling2D
np.random.seed(123)  # for reproducibility
import scipy.stats as stats
from statsmodels.miscmodels.ordinal_model import OrderedModel

#close warnings in python
import warnings
warnings.filterwarnings("ignore")

from numpy import mean
from numpy import std
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier

2# import dataset
df = pd.read_excel (r'C:\Users\Huseyin\Desktop\article\MAKALE.xlsx', sheet_name='sheet')
df.head()

3# exclude some columns with drop method
veri = df.drop(["CVE (cybersecurity vulnerabilities enumeration)","CVE","Number","Year", "SCORE", "CVSS
(Common Vulnerability Scoring System)", "CWE", "CWE (Common Weakness Enumeration)", "observation"],
axis = 1) # axis = 1 sütunları çıkarmak anlamında
veri.head()

4#determine dependent and independent varables
y = veri[["Category"]]
x = veri[["Attack Complexity", "Privileges Required", "User Interaction", "Scope", "Confidentiality",
"Integrity", "Availability"]]
x.info()
y.info()

5#lineer regression model
regressor = LinearRegression()
regressor.fit(x, y)
r_sq = regressor.score(x, y)
print('coefficient of determination:', r_sq)
print('intercept:', regressor.intercept_)
print('slope:', regressor.coef_)

6#ordered regression model
mod_prob = OrderedModel(veri['Category'],
                        veri[["Attack Complexity", "Privileges Required", "User Interaction", "Scope", "Confidentiality",
"Integrity", "Availability"]],
                        distr='probit')
res_prob = mod_prob.fit(method='bfgs')
res_prob.summary()

7# Decion Tree Best
reportrandom =[]
reportdecision =[]
reportknn =[]
for i in range (150):
    train_data = veri.sample(frac =.80)
    y_train = train_data[["Category"]]
    x_train = train_data[["Attack Complexity", "Privileges Required", "User Interaction", "Scope",
"Confidentiality", "Integrity", "Availability"]]
    test_data = veri.sample(frac =.20)
    y_test = test_data[["Category"]]
    x_test = test_data[["Attack Complexity", "Privileges Required", "User Interaction", "Scope",
"Confidentiality", "Integrity", "Availability"]]
    model = RandomForestClassifier()
    model.fit(x_train, y_train)
    ypred = model.predict(x_test)
    report = accuracy_score(ypred, y_test)
    reportrandom.append(report)

    model = DecisionTreeClassifier()
    model.fit(x_train, y_train)
    ypred = model.predict(x_test)
    report = accuracy_score(ypred, y_test)
    reportdecision.append(report)
```



```
    model = KNeighborsClassifier(n_neighbors=7)
    model.fit(x_train, y_train)
    ypred = model.predict(x_test)
    report = accuracy_score(ypred, y_test)
    reportknn.append(report)
print("Random Fores accuracy rate:  ", np.mean(reportrandom))
print("DecisionTree accuracy rate:  ", np.mean(reportdecision))
print("KNN accuracy rate:  ", np.mean(reportknn))
```

8# Best Model-KNN

```
reportrandom =[]
reportdecision =[]
reportknn =[]
for i in range (500):
    y = veri[["Category"]]
    x = veri[["Attack Complexity", "Privileges Required", "User Interaction", "Scope", "Confidentiality",
"Integrity", "Availability"]]
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0, stratify=y)
    model = RandomForestClassifier()
    model.fit(x_train, y_train)
    ypred = model.predict(x_test)
    report = accuracy_score(ypred, y_test)
    reportrandom.append(report)

    model = DecisionTreeClassifier()
    model.fit(x_train, y_train)
    ypred = model.predict(x_test)
    report = accuracy_score(ypred, y_test)
    reportdecision.append(report)

    model = KNeighborsClassifier(n_neighbors=7)
    model.fit(x_train, y_train)
    ypred = model.predict(x_test)
    report = accuracy_score(ypred, y_test)
    reportknn.append(report)
print("Random Fores accuracy rate:  ", np.mean(reportrandom))
print("DecisionTree accuracy rate:  ", np.mean(reportdecision))
print("KNN accuracy rate:  ", np.mean(reportknn))
```

9# Random Forest

```
    reportrandom =[]
    reportdecision =[]
    reportknn =[]
    for i in range (500):
        y = veri[["Category"]]
        x = veri[["Attack Complexity", "Privileges Required", "User Interaction", "Scope", "Confidentiality",
    "Integrity", "Availability"]]
        x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=0, stratify=y)
        model = RandomForestClassifier()
        model.fit(x_train, y_train)
        ypred = model.predict(x_test)
        report = accuracy_score(ypred, y_test)
        reportrandom.append(report)

        model = DecisionTreeClassifier()
        model.fit(x_train, y_train)
        ypred = model.predict(x_test)
        report = accuracy_score(ypred, y_test)
        reportdecision.append(report)

        model = KNeighborsClassifier(n_neighbors=7)
        model.fit(x_train, y_train)
        ypred = model.predict(x_test)
        report = accuracy_score(ypred, y_test)
        reportknn.append(report)
    print("Random Fores accuracy rate:  ", np.mean(reportrandom))
    print("DecisionTree accuracy rate:  ", np.mean(reportdecision))
    print("KNN accuracy rate:  ", np.mean(reportknn))
```

10 # Define model architecture with ML-Keras

```
    result=[]
    model = tf.keras.Sequential([
    keras.layers.Dense(units=250, input_shape=[7],
                    activation=tf.nn.relu),
    keras.layers.Dense(units=100,
                    activation=tf.nn.relu), # activation='relu'
    keras.layers.Dense(units=50, activation=tf.nn.relu),
    keras.layers.Dense(units=10, activation='softmax')
    ])

    for i in range (100):
        y = veri[["Category"]]
        x = veri[["Attack Complexity", "Privileges Required", "User Interaction", "Scope", "Confidentiality",
    "Integrity", "Availability"]]
        x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, stratify=y)

        model.compile(
        optimizer='adam',
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy']
        )
        model.fit(x_train, y_train, epochs=250, verbose=0)
        result.append(model.evaluate(x_test, y_test, batch_size=10, verbose=0))
    df = pd.DataFrame (result)
```

Funding No funding was received to assist with the preparation of this manuscript.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

Abu-Aisheh Z, Raveaux R, Ramel JY (2020) Efficient k-nearest neighbors search in graph space. Pattern Recognit Lett 134:77–86

Angel D (2022) Application of graph domination to defend medical information networks against cyber threats. J Ambient Intell Humaniz Comput 13:3765–3770

Anikin IV (2017) Using fuzzy logic for vulnerability assessment in telecommunication network. In: International conference on industrial engineering, applications and manufacturing (ICIEAM)

Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor 18(2):1153–1176

Chen H, Chillotti I, Dong Y, Poburinnaya O, Razenshteyn I, Riazi MS (2020) SANNS: scaling up secure approximate k-nearest neighbors search. In: Proceedings of the 29th USENIX security symposium, 2020, pp 2111–2128

Chomboon K, Chujai P, Teerarassamee P, Kerdprasop K, Kerdprasop N (2015) An empirical study of distance metrics for k-nearest neighbor algorithm. In: Proceedings of the 3rd international conference on industrial application engineering, pp 280–285

Coventry L, Branley D (2018) Cybersecurity in healthcare: a narrative review of trends, threats and ways. Maturitas 113:48–52

Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory IT-13(1):21–27

Desruelle P, Baldini G, Barboni M, Bono F, Delipetrev B, Duch Brown N, Fernandez Macias E, Gkoumas K, Joossens E, Kalpaka A, Nepelski D, Nunes de Lima MV, Pagano A, Prettico G, Sanchez I, Sobolewski M, Triaille J-P, Tsakalidis A, Urzi Brancati MC (2019) Digital transformation in transport, construction, energy, government and public administration, EUR 29782 EN. Publications Office of the European Union, Luxembourg

Dondo MG (2008) A vulnerability prioritization system using a fuzzy risk analysis approach. Springer US, Boston

Durbin M, Wonders M, Flaska M, Lintereur AT (2020) K-nearest neighbors regression for the discrimination of gamma rays and neutrons in organic scintillators. Nucl Instrum Methods Phys Re Sect A Accelerators Spectrom Detect Assoc Equip 987:164826

Durkota K, Lisý V, Bošanský B, Kiekintveld C, Pěchouček M (2019) Hardening networks against strategic attackers using attack graph games. Comput Secur 87:101578

Ertem M, Bier VM (2021) A stochastic network-interdiction model for cyber security. In: 5th international symposium on multidisciplinary studies and innovative technologies (ISMSIT), 2021, pp 171–176

FIRST.org. (2022) Common vulnerability scoring system version 3.1: specification document. https://www.first.org/cvss/specification-document. Accessed 19 June 2022

Gencer K, Başçiftçi F (2020) The fuzzy common vulnerability scoring system (F-CVSS) based on a least squares approach with fuzzy logistic regression. Egypt Inform J 22(2):145–153

Gou J, Qiu W, Yi Z, Xu Y, Mao Q, Zhan Y (2019) A local mean representation-based k-nearest neighbor classifier. ACM Trans Intell Syst Technol 10(3):1–25

Islam S, Papastergiou S, Kalogeraki EM, Kioskli K (2022) Cyberattack path generation and prioritisation for securing healthcare systems. Appl Sci 12(9):4443

Kaun C, Jhanjhi NZ, Goh WW, Sukumaran S (2021) Implementation of decision tree algorithm to classify knowledge quality in a knowledge intensive system. In: 14th EURECA 2020 MATEC web of conferences, vol 335, p 04002.

Kaynar K (2016) A taxonomy for attack graph generation and usage in network security. J Inf Secur Appl 29:27–56

Keramati M, Akbari A (2013) CVSS-based security metrics for quantitative analysis of attack graphs. In ICCKE 2013 IEEE (pp. 178–183)

Khazaei A, Ghasemzadeh M, Derhami V (2016) An automatic method for CVSS score prediction using vulnerabilities description. J Intell Fuzzy Syst 30:89–96

Lallie HS, Debattista K, Bal J (2020) A review of attack graph and attack tree visual syntax in cyber security. Comput Sci Rev 35:100219

Li BL, Yu SW, Lu Q (2003) An improved k-nearest neighbour algorithm for text categorization. In: Proceedings of the international conference on computing processing oriental language, Jan 1, pp 469–475

Lorenzo F, Añorga SJ, Arrizabalaga S (2020) A survey of IIoT protocols: a measure of vulnerability risk analysis based on CVSS. ACM Comput Surv (CSUR) 53(2):1–53

Majeed A, Lee S (2021) Towards privacy paradigm shift due to the pandemic: a brief perspective. Inventions 6(2):24

National Institute of Standards and Technology (2022). Vulnerabilities. https://nvd.nist.gov/vuln. Accessed 19 June 2022

Pollini A, Callari TC, Tedeschi A, Ruscio D, Save L, Chiarugi F, Guerri D (2022) Leveraging human factors in cybersecurity: an integrated methodological approach. Cogn Technol Work 24(2):371–390

Schonlau M, Zou RY (2020) The random forest algorithm for statistical learning. Stand Genom Sci 20(1):3–29

Sheyner O, Haines J, Jha S, Lippmann R, Wing JM (2002) Automated generation and analysis of attack graphs. In: Proceedings 2002 IEEE symposium on security and privacy. IEEE, pp 273–284

Teal, K. (2020). Cybercrime tactics and techniques: COVID-19 sends attackers into overdrive, channel futures. https://www.channelfutures.com/mssp-insider/cybercrime-tactics-and-techniques-covid-19-sends-attackers-into-overdrive. Accessed 19 June 2022

Wang T, Lv Q, Hu B, Sun D (2020) CVSS-based multi-factor dynamic risk assessment model for network system. In: IEEE 10th international conference on electronics information and emergency communication (ICEIEC)

Wu C, Wen T, Zhang Y (2019) A revised CVSS-based system to improve the dispersion of vulnerability risk scores. Sci China Inf Sci 62(3):039102

Yilmaz M, Ünözkan H (2015) A study on mathematical model of determining three Istanbul football clubs winning or losing. Niğde Univ J Phys Educ Sport Sci 9(1):94–104

Zhang S (2011) Shell-neighbor method and its application in missing data imputation. Appl Intell 35(1):123–133

Zhang H, Lou F, Fu Y, Tian Z (2017) A conditional probability computation method for vulnerability exploitation based on CVSS. In: IEEE second international conference on data science in cyberspace (DSC)

Zheng L, Huang H, Zhu C, Zhang K (2020) A tensor-based k-nearest neighbors method for traffic speed prediction under data missing. Transportmetr B Transp Dyn 8(1):182–199

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.