ZERO-DAY EXPLOIT NETWORK LEAKAGE PREDICTION USING MACHINE LEARNING

Mrs. JACKULIN ASHA, M.E., Assistant Professor,

Department of Computer Science and Engineering

Mr. D. NARESH BABU B.E, Student of Computer Science Engineering

Mr. V. JAYAKUMAR B.E, Student of Computer science and Engineering

St. Joseph College of Engineering, Sriperumbudur, Chennai.

Abstract

Wireless sensor network has attracted significant attention in research and development due to its tremendous applications in medical, military and defense, medical, environmental, industrial, infrastructure protection, and commercial applications to enable to interact with each other controlled remotely. A Wireless Sensor Network (WSN) has wide applications such as environmental monitoring and tracking of the target nodes for communication. The sensor nodes are equipped with wireless interfaces used for communication between the nodes and another network. Wireless Sensor Network suffers from many constraints that make security a primary challenge. When the sensor node is deployed in a communication environment unattended, the nodes are vulnerable to various attacks. The analysis of dataset by supervised machine learning technique (SMLT) to capture several information's like, variable identification, univariate analysis, bivariate and multivariate analysis, missing value treatments etc. A comparative study between machine learning algorithms had been carried out in order to determine which algorithm is the most accurate in predicting the type WSN attacks. The results show that the effectiveness of the proposed machine learning algorithm technique can be compared with best accuracy, precision, Recall, F1 Score, Sensitivity, and Specificity.

Introduction

The most devastating and complicated attack in a wireless sensor network is the Wormhole attack. In this attack, the attacker keeps track of the packets and makes a tunnel with other nodes of different communication networks, and thus the attacker passes the packets through this tunnel. And the outsider attack can be prevented by authentication and encryption techniques by launching a Sybil attack on the sensor network. In WSN the routing protocols in network has a unique identity. The figure demonstrates Sybil attack where an attacker node is present with multiple identities.

Literature Survey

The paper "Stroke Risk Prediction Model based on Demographic Data." By Teerapat Kansadub, Sotarat Thammaboosade kiattisin in 2015, provide the development of model for prediction based on the demographic data of the patients. This study aim to compare accuracy, false positive (FP), false negative (FN), and area under ROC Curve (AUC) resulted from three methods among Decision tree, Naïve Bayes, and Neural Network and then converted to rule. The best rule is selected for the benefits of population who have risk in stroke.

The paper "Prediction of Stroke using Data Mining Classification Techniques." By Ohoud Almadani, Riyad Alshammari in 2018, have considered Several assessments and prediction models, Decision Tree, Naive Bayes and Neural Network, showed acceptable accuracy in identifying stroke-prone patients. This project hence helps to predict the stroke risk using prediction model and provide personalized warning and the lifestyle correction message through a web application. By doing so, it urges medical users to strengthen the motivation of health management and induce changes in their health behaviors.

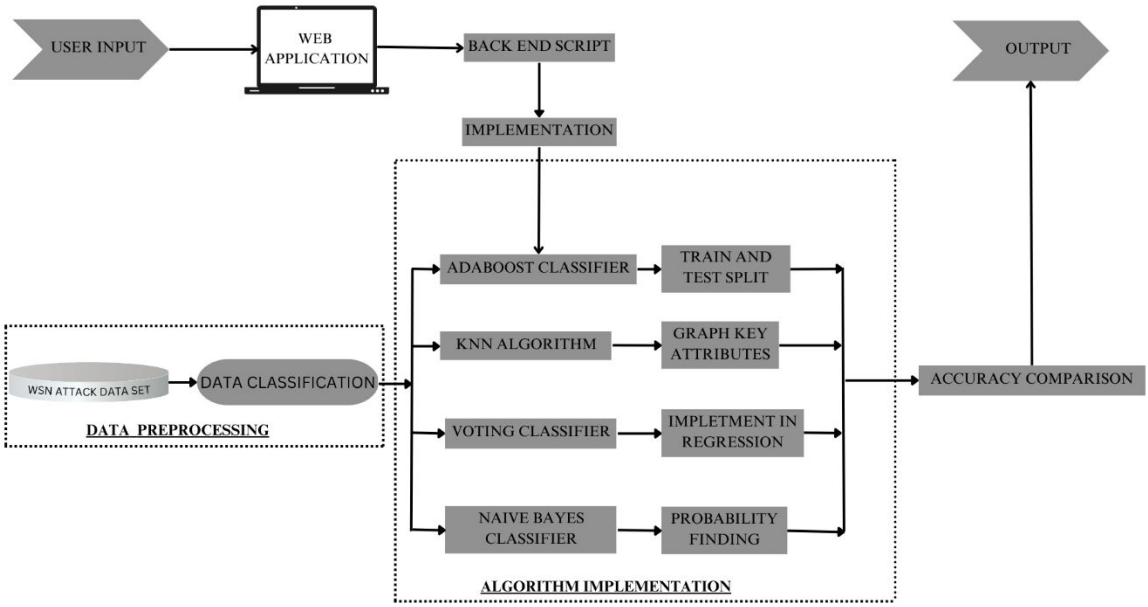
The paper" Prediction of Stroke Using Machine Learning." By Kunder Akash Mahesh, Srikanth S, Shashank H N in 2020, helps to predict the stroke risk using data mining classification techniques.

The main objectives of this research are twofold: i) Use data mining techniques to predict patient at risk of developing stroke; and ii) Find the patient with who has higher chances to develop stroke.

System Design

This analysis aims to observe which features are most helpful in predicting the WSN of BLACKHOLE, FLOODING, GRAYHOLE, NORMAL and SCHEDULING to see the general trends that may help us in model selection and hyper parameter selection. To achieve used machine learning classification methods to fit a function that can predict the discrete class of new input.

WSN is one of the major factors in our major domain. The most devastating and complicated attack in a wireless sensor network is the Wormhole attack. In this attack, the attacker keeps track of the packets and makes a tunnel with other nodes of different communication networks, and thus the attacker passes the packets through this tunnel. So this project can easily find out the Attack.



Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning models and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

IMPLEMENTATION

Data Pre-Processing

```
import pandas as p

import numpy as n

import warnings warnings.filterwarnings('ignore')

data = p.read_csv('WSN_Dataset.csv')

data.head()

data.shape
```

```
df = data.dropna()
df.shape
df.isnull().sum()
df.columns
df.describe()
df.Is_CH.unique() '
df.JOIN_R.unique()
df.send_code.unique()
p.crosstab(df.Attack_type,df.send_code)
p.Categorical(df['Is_CH']).describe()
p.Categorical(df['send_code']).describe()
print("Time: ", sorted(df['Time'].unique()))
print("ADV_R: ", sorted(df['ADV_R'].unique()))
df['Attack_type'].value_counts()
df.duplicated()
df.duplicated().sum()
df.columns
print("Minimum value of Dist_To_CH is:", df.Dist_To_CH.min())
print("Maximum value of Dist_To_CH is:", df.Dist_To_CH.max())
df.duplicated().sum()
df.columns
print("Minimum value of Dist_To_CH is:", df.Dist_To_CH.min())
print("Maximum value of Dist_To_CH is:", df.Dist_To_CH.max())
df.info()
df['Attack_type'].unique()
from sklearn.preprocessing import LabelEncoder var_
```

```
mod = ['Attack_type'] le = LabelEncoder()
for i in var_mod:
        df[i] = le.fit_transform(df[i]).astype(int)
df['Attack_type'].unique()
df.tail()
```

Gaussian Naive Bayes

```
#import library packages
import pandas as p
import warnings warnings.filterwarnings('ignore')
#Load given dataset
data = p.read_csv("WSN_Dataset.csv")
df = data.drop_duplicates()
del df['id']
df.info()
df['Attack_type'].unique()
from sklearn.preprocessing
import LabelEncoder col = ['Attack_type']
label = LabelEncoder()
for i in col:
        df[i] = label.fit_transform(df[i]).astype(int)
df['Attack_type'].unique()
inputs = df.drop(labels='Attack_type', axis=1)
output = df.loc[:,'Attack_type']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs, output, test_size=0.3, random
print("Number of Training Dataset: ", len(X_train))
print("Number of Testing Dataset: ", len(X_test))
print("Total Number of Dataset: ", len(X_train)+len(X_test))
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, plot
gnb = GaussianNB()
```

```
gnb.fit(X_train,y_train)
predicted_nb = gnb.predict(X_test)
accuracy = accuracy_score(y_test,predicted_nb)
print('Accuracy of Gaussian Naive Bayes is: ',accuracy*100)
cr = classification_report(y_test,predicted_nb)
print('Classification report\n----------------------\n',cr)
cm = confusion_matrix(y_test,predicted_nb)
print('Confusion matrix\n----------------\n',cm)
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(7,7))
plot_confusion_matrix(gnb, X_test, y_test, ax=ax)
plt.title('Confusion Matrix of Gaussian Naive Bayes\n')
plt.show()
DF = p.DataFrame()
DF["y_test"] = y_test
DF["predicted"] = predicted_nb
DF.reset_index(inplace=True)
plt.figure(figsize=(20, 5))
plt.plot(DF["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(DF["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()
```

AdaBoost Classifier

```
#import library packages
import pandas as p
import warnings warnings.filterwarnings('ignore')
#Load given dataset
data = p.read_csv("WSN_Dataset.csv")
df = data.dropna()
del df['id'] df.info()
df['Attack_type'].unique()
from sklearn.preprocessing import LabelEncoder
```

```
col = ['Attack_type'] label = LabelEncoder()
for i in col:
        df[i] = label.fit_transform(df[i]).astype(int)
df['Attack_type'].unique()
inputs = df.drop(labels='Attack_type', axis=1)
output = df.loc[:,'Attack_type']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs, output, test_size=0.3, random
print("Number of Training Dataset: ", len(X_train))
print("Number of Testing Dataset: ", len(X_test))
print("Total Number of Dataset: ", len(X_train)+len(X_test))
from sklearn.ensemble
import AdaBoostClassifier from sklearn.metrics
import confusion_matrix, classification_report, accuracy_score, plot
ab = AdaBoostClassifier()
ab.fit(X_train,y_train)
predicted_ab = ab.predict(X_test)
accuracy = accuracy_score(y_test,predicted_ab)
print('Accuracy of Ada Boost Classifier is: ',accuracy*100)
cr = classification_report(y_test,predicted_ab)
print('Classification report\n----------\n',cr)
cm = confusion_matrix(y_test,predicted_ab)
print('Confusion matrix\n----------\n',cm)
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(7,7))
plot_confusion_matrix(ab, X_test, y_test, ax=ax)
plt.title('Confusion Matrix of AdaBoost Classifier \n')
plt.show()
DF = p.DataFrame()
DF["y_test"] = y_test
DF["predicted"] = predicted_ab
DF.reset_index(inplace=True)
```

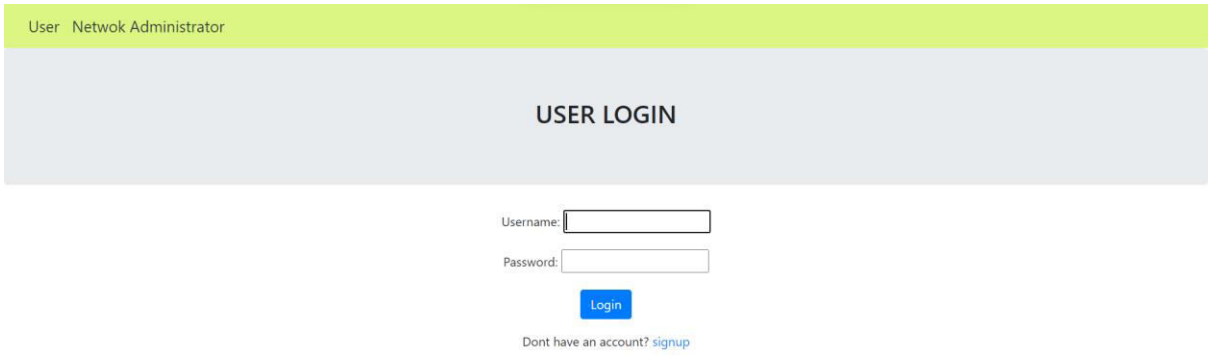
```
plt.figure(figsize=(20, 5))
plt.plot(DF["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(DF["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()
```

K-Nearest Neighbors

```
#import library packages
import pandas as p
import warnings warnings.filterwarnings('ignore')
#Load given dataset
data = p.read_csv("WSN_Dataset.csv")
df = data.drop_duplicates()
df
del df['id']
df.info()
df['Attack_type'].unique()
from sklearn.preprocessing
import LabelEncoder col = ['Attack_type']
label = LabelEncoder()
for i in col:
        df[i] = label.fit_transform(df[i]).astype(int)
df['Attack_type'].unique()
df
inputs = df.drop(labels='Attack_type', axis=1)
output = df.loc[:,'Attack_type']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs, output, test_size=0.3, random
print("Number of Training Dataset: ", len(X_train))
print("Number of Testing Dataset: ", len(X_test))
print("Total Number of Dataset: ", len(X_train)+len(X_test))
from sklearn.neighbors import KNeighborsClassifier from sklearn.metrics
import confusion_matrix, classification_report, accuracy_score, plot
```

```
knn = KNeighborsClassifier() knn.fit(X_train,y_train)
predicted_knn = knn.predict(X_test)
accuracy = accuracy_score(y_test,predicted_knn)
print('Accuracy of K-Nearest Neighbors is: ',accuracy*100)
cr = classification_report(y_test,predicted_knn)
print('Classification report\n----------\n',cr)
cm = confusion_matrix(y_test,predicted_knn)
print('Confusion matrix\n----------\n',cm)
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(7,7))
plot_confusion_matrix(knn, X_test, y_test, ax=ax)
plt.title('Confusion Matrix of K-Nearest Neighbors')
plt.show()
DF = p.DataFrame() DF["y_test"] = y_test
DF["predicted"] = predicted_knn DF.reset_index(inplace=True)
plt.figure(figsize=(20, 5))
plt.plot(DF["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(DF["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()
import joblib
joblib.dump(knn,'model.pkl')
```

SNAPSHOTS


User
Netwok Administrator
USER LOGIN
Username:
Password:
Login
Dont have an account? signup


Logout
Enter Details
Feedback
WIRELESSS SENSOR NETWORK ATTACK PREDICTION
Enter the Detail
Time
0.86
Is_CH
0.71
who_CH
0.43
Dist_To_CH
0.51


ADV_S
0.66
ADV_R
0.87
JOIN_S
0.54
JOIN_R
0.93
SCH_S
0.6
SCH_R
0.48
Rank


Rank
0.65
DATA_S
0.83
DATA_R
1.04
Data_Sent_To_BS
0.21
dist_CH_To_BS
0.42
send_code
0.42

Logout

All Last

Time	Is_CH	who_CH	Dist_To_CH	ADV_S	ADV_R	JOIN_S	JOIN_R	SCH_S	SCH_R	Rank	DATA_S	DATA_R	Data_Sent_To_BS	dist_CH_To_BS	send_code	Expaned_Energy	Attack_type
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Normal
4	5	56	45	4	5	2	4	5	54	5	1	2	52	5	2	5	Normal
1	2	3	4	5	6	8	7	8	5	4	5	2	5	2	5	2	Black hole
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	8	Black hole
1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	74	8	Normal
159	753	654	852	159	753	654	852	159	654	755	623	5145	85	25	6	32	Flooding

Conclusion

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score will be find out by comparing each algorithm with type of all WSN Attacks for future prediction results by finding best connections. This brings some of the following insights about diagnose the network attack of

each new connection. To presented a prediction model with the aid of artificial intelligence to improve over human accuracy and provide with the scope of early detection. It can be inferred from this model that, area analysis and use of machine learning technique is useful in developing prediction models that can helps to network sectors reduce the long process of diagnosis and eradicate any human error.

FUTURE ENHANCEMENTS:

Network sector want to automate and detecting the attacks of packet transfers from eligibility process (real time) based on the connection detail. To automate this process by show the prediction result in web application or desktop application at cloud. To optimize the work to implement in Artificial Intelligence environment.

REFERENCES:

[1] Lei Yang, Qing "**Combined dual-prediction based data fusion and enhanced leak detection and isolation method for WSN pipeline monitoring system**"

[2] Lakshmana Kumar Ramasamy, Firoz khan k. P. "**Wireless Sensor Networks (WSNs) are broadly applied for various applications in tracking and surveillance due to their ease of use and other distinctive characteristics compelled by real-time cooperation among the sensor nodes**"

[3] Abdollah Kavousi-Fard, Wencong Su, Tao Jin "**This paper proposes an accurate secured framework to detect and stop data integrity attacks in wireless sensors networks in micro grids**"

[4] B.J Santhosh Kumar, Somnath Sinha "**An Intrusion Detection and Prevention System against DOS Attacks for Internet-Integrated WSN** "

[5] M. Keerthika, D. Shanmugapriya "**A Systematic Survey on Various Distributed Denial of Service (DDoS) Attacks in Wireless Sensor Networks (WSN)**"

[6] S. Uma Maheswari, N. S. Usha, E. A. Mary Anita, K. Ramaya Devi "**A novel robust routing protocol RAEED to avoid DoS attacks in WSN**"

[7] Divya Acharya, ShubhLakshmi Agrwal, Pankaj Sharma, Sandeep Kumar Gupta "**Performance analysis of detection technique for select forwarding attack on WSN** "

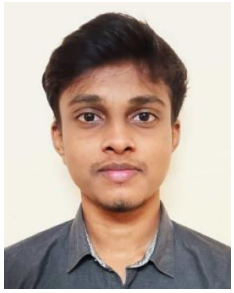
AUTHOR 1

Mrs. G.S.JACKULIN ASHA M.E. is a Department of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. She has completed her M.E, in Anna University Computer Science and Engineering in 2013 from Chennai, Tamil Nadu. She has done her B.E, CSE in Anna University from Nagercoil in the year 2011. Mrs. G.S.JACKULIN ASHA has 10 years of teaching experience and has 10 publications in International Journals and conference.

AUTHOR 2

Mr. Naresh Babu D BE Student of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. I had attended many Workshops, Seminars in Java programming, data Analytics. I completed the python full stack course. I got placed in Reputed Companies like, Aveon Info system, Q Spider and some respected companies.

AUTHOR 3



Mr. Jayakumar VBE Student of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. I had attended many Workshops, Seminars in Java programming, data Analytics.