

PROJECT TITLE

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

ABHIPSA SAHOO [RA2011003010346]

ARUSHI GUPTA [RA2011003010392]

ADARSH MAMGAIN

[RA2011003010381]

Under the guidance of

SELVARAJ P.

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that the Mini project report titled “**PROJECT TITLE**” is the bona fide work of **ABHIPSA SAHOO [RA2011003010346]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

GUIDE NAME

GUIDE

Assistant Professor

Department of Computing
Technologies

SIGNATURE

Dr. M. Pushpalatha

HEAD OF THE DEPARTMENT

Professor & Head

Department of Computing
Technologies

ABSTRACT

This report examines the use of AI chatbots for cybersecurity purposes. With the increase in cyber threats, organizations are seeking innovative ways to mitigate the risk of data breaches and protect sensitive information. AI chatbots have emerged as a promising technology to enhance cybersecurity measures. The report analyzes the various applications of AI chatbots in cybersecurity, including threat detection, incident response, and user authentication. It also discusses the benefits of using AI chatbots, such as faster response times, improved accuracy, and reduced costs. Furthermore, the report highlights the challenges and limitations associated with AI chatbots in cybersecurity, such as data privacy concerns and the potential for bias in decision-making. Overall, the report concludes that AI chatbots have the potential to revolutionize cybersecurity and improve the overall security posture of organizations.

In addition to discussing the applications, benefits, challenges, and limitations of AI chatbots for cybersecurity, this report also delves into the technical aspects of implementing such technology. It covers the different types of AI chatbots, such as rule-based and machine learning-based, and how they can be customized to meet specific cybersecurity needs. The report also discusses the importance of data quality and training data for AI chatbots, as well as the role of natural language processing (NLP) in enabling effective communication between chatbots and users.

Moreover, the report examines the regulatory and ethical considerations surrounding the use of AI chatbots for cybersecurity. It explores the legal implications of using AI chatbots for data protection and compliance with industry standards, such as the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS). It also discusses ethical concerns, such as the potential for chatbots to be used for social engineering and the need for transparency in AI decision-making.

Finally, the report provides case studies of organizations that have successfully implemented AI chatbots for cybersecurity. These case studies showcase the benefits of using AI chatbots, such as increased efficiency and accuracy, and how they have helped organizations detect and respond to cyber threats. The report concludes with recommendations for organizations looking to implement

AI chatbots for cybersecurity, such as conducting a thorough risk assessment and ensuring transparency and accountability in AI decision-making.

TABLE OF CONTENTS

ABSTRACT	i
TABLE OF CONTENTS	ii
LIST OF FIGURES	v
ABBREVIATIONS	vi
1 INTRODUCTION	7
2 LITERATURE SURVEY	8
3 SYSTEM ARCHITECTURE AND DESIGN	9
3.1 Architecture diagram of proposed IoT based smart agriculture project	9
3.2 Description of Module and components	10
4 METHODOLOGY	14
4.1 Methodological Steps	14
5 CODING AND TESTING	15
	22
7 CONCLUSION AND FUTURE ENHANCEMENT	23
7.1 Conclusion	
7.2 Future Enhancement	
REFERENCES	24

ABBREVIATIONS

NLP - Natural Language Processing

AI - Artificial Intelligence

IDS - Intrusion Detection System

SIEM - Security information and
event management

GDPR - General Data Protection Regulation

INTRODUCTION

In today's digital age, cyber security is a critical concern for organizations of all sizes. With the increasing number of cyber threats, businesses must ensure that they have effective security measures in place to protect their sensitive information and digital assets. However, maintaining a high level of cyber security can be challenging, as the threat landscape is constantly evolving and becoming more sophisticated. In response to these challenges, artificial intelligence (AI) has emerged as a promising solution to help organizations enhance their cyber security measures. AI-based chatbots are intelligent software programs that can interact with users through natural language processing (NLP) and machine learning algorithms. These chatbots can understand and respond to user queries related to cyber security, providing real-time alerts for potential threats, and automating routine security tasks. The AI chatbot in the field of cyber security is an innovative solution that leverages advanced technologies to provide timely and accurate information to users. The chatbot can analyze and interpret large amounts of data, detect anomalies, and alert security personnel to potential threats in real-time. It can also assist in vulnerability assessment and offer advice on best practices to mitigate potential risks.

Moreover, the AI chatbot can help organizations to reduce the response time to security incidents, providing a more efficient and effective approach to cyber security. It can continuously monitor network activities, identify potential security breaches, and take proactive measures to prevent or mitigate the impact of an attack.

In summary, the AI chatbot in the field of cyber security is a valuable tool that can assist organizations in strengthening their cyber security posture and protecting their assets against emerging threats. This paper will provide an overview of the key features and benefits of the AI chatbot, as well as its potential applications in different industries.

LITERATURE SURVEY

INTRODUCTION:

As the world becomes increasingly digitized, cybersecurity threats continue to pose a significant risk to organizations' digital assets. AI chatbots have emerged as a promising solution to enhance cybersecurity measures. In this literature survey, we explore the existing literature on AI chatbots designed for cybersecurity and their potential to help organizations safeguard their digital assets from potential cyber threats.

Applications of AI chatbots in cybersecurity:

AI chatbots can be used for various cybersecurity applications, including threat detection, incident response, and user authentication. Chatbots can detect suspicious activity and anomalous behavior in real-time, enabling organizations to respond quickly to potential threats. Chatbots can also provide automated responses to common cybersecurity queries and educate users on best practices for cybersecurity.

Benefits of AI chatbots for cybersecurity:

AI chatbots can provide numerous benefits for organizations' cybersecurity, including faster response times, improved accuracy, and reduced costs. Chatbots can operate 24/7, providing around-the-clock protection against cyber threats. They can also be trained to learn from previous incidents, allowing for continuous improvement in detecting and responding to potential threats.

Challenges and limitations of AI chatbots for cybersecurity:

While AI chatbots have many benefits for cybersecurity, there are also challenges and limitations to their implementation. One of the main challenges is ensuring that chatbots are trained on high-quality data to prevent bias and false positives. Additionally, chatbots must be designed with user privacy in mind, and data protection regulations must be adhered to. Finally, chatbots are not a complete solution to cybersecurity and must be used in conjunction with other cybersecurity measures.

Technical aspects of AI chatbots for cybersecurity:

There are different types of AI chatbots, including rule-based and machine learning-based. Rule-based chatbots follow predefined rules, while machine learning-based chatbots use algorithms to learn from data and improve over time. Natural language processing (NLP) is also a crucial component of chatbots, enabling them to communicate effectively with users.

Regulatory and ethical considerations of AI chatbots for cybersecurity:

There are regulatory and ethical considerations when implementing AI chatbots for cybersecurity. Organizations must comply with data protection regulations, such as the GDPR, and ensure that user privacy is protected. Additionally, there must be transparency in AI decision-making, and chatbots must not be used for malicious purposes.

Case studies of AI chatbots in cybersecurity:

Several organizations have successfully implemented AI chatbots for cybersecurity. For example, one organization used a chatbot to detect and respond to phishing attacks, reducing the response time from hours to minutes. Another organization implemented a chatbot to educate users on cybersecurity best practices, resulting in a 90% reduction in user errors.

Conclusion:

In conclusion, AI chatbots have the potential to revolutionize cybersecurity and help organizations safeguard their digital assets from potential cyber threats. However, their implementation must be carefully considered, with attention paid to technical, regulatory, and ethical considerations. Future research should continue to explore the potential of AI chatbots for cybersecurity and identify ways to improve their effectiveness in detecting and responding to cyber threats.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

The architecture for the AI chatbot in the field of cyber security is based on a multi-layered approach, consisting of several key components:

User Interface: The chatbot's user interface allows users to interact with the chatbot through natural language processing (NLP). Users can ask questions related to cyber security and receive responses in real-time.

NLP Engine: The NLP engine processes user queries and extracts relevant information from them. It uses machine learning algorithms to understand user intent and provide accurate responses.

Knowledge Base: The knowledge base is a repository of information related to cyber security. It includes best practices, vulnerability assessments, and potential threats, which are continuously updated by security experts.

Multilinguality : The AI chatbot's multilingual capabilities in Hindi, English, and Tamil enable personalized and efficient user experiences, improving cyber security measures.

Alerting System: The alerting system sends notifications to security personnel when potential threats are detected, allowing for a timely response to security incidents.

Integration with Existing Systems: The chatbot can be integrated with existing security systems and workflows, allowing for seamless collaboration and automation of routine security tasks.

Overall, the proposed architecture for the AI chatbot in the field of cyber security is designed to provide organizations with an efficient and effective approach to enhancing their cyber security measures. The multi-layered approach ensures that the chatbot can provide accurate and timely responses to user queries while also monitoring network activities and alerting security personnel to potential threats.

CHAPTER 4

METHODOLOGY

The methodology for developing an AI chatbot for cybersecurity involves several key steps:

1. Identify the use case: The first step is to determine the specific use case for the chatbot. This could include threat detection, incident response, user authentication, or a combination of these.
2. Gather data: Once the use case is identified, data must be gathered to train the chatbot. This data may include logs, network traffic, and other security-related information.
3. Pre-process the data: Before the data can be used to train the chatbot, it must be cleaned, formatted, and pre-processed. This may involve removing irrelevant data, transforming data into a usable format, and handling missing values.
4. Train the chatbot: The pre-processed data is then used to train the chatbot. Depending on the type of chatbot being developed, this may involve rule-based programming or machine learning algorithms.
5. Test the chatbot: Once the chatbot has been trained, it must be tested to ensure that it is functioning correctly. This involves running it through a series of scenarios to see how it responds.
6. Deploy the chatbot: Once the chatbot has been tested, it can be deployed in a production environment. This may involve integrating it with existing security tools and systems.
7. Monitor the chatbot: Finally, the chatbot must be monitored on an ongoing basis to ensure that it is functioning as intended. This may involve collecting feedback from users and making adjustments to improve its performance.

Overall, the methodology for developing an AI chatbot for cybersecurity requires a combination of technical expertise in machine learning, natural language processing, and cybersecurity, as well as an understanding of the specific needs and challenges of the organization in question. By following a systematic approach to development, organizations can create a chatbot that is effective at detecting and responding to cyber threats in real-time.

CHAPTER 5

CODING AND TESTING

```
import nltk
nltk.download("punkt")
from nltk.stem.lancaster import LancasterStemmer
stemmer = LancasterStemmer()
import tensorflow as tf #version 1.13.2
import numpy as np
import tflearn #version 0.3.2
import random
import json
with open("intents.json") as json_data:
    intents = json.load(json_data)
words=[]
documents = []
classes = []

# This list will be used for ignoring all unwanted punctuation marks.
ignore = ["?"]

# Starting a loop through each intent in intents["patterns"]
for intent in intents["intents"]:
    for pattern in intent["patterns"]:

5

# tokenizing each and every word in the sentence by using word tokenizer and storing in w
w = nltk.word_tokenize(pattern)
```

```

#print(w)

# Adding tokenized words to words empty list that we created
words.extend(w)

#print(words)


# Adding words to documents with tag given in intents file
documents.append((w, intent["tag"]))

#print(documents)


# Adding only tag to our classes list
if intent["tag"] not in classes:
    classes.append(intent["tag"]) #If tag is not present in classes[] then it will append into it.
#print(classes)


words = [stemmer.stem(w.lower()) for w in words if w not in ignore]
words = sorted(list(set(words))) #Removing Duplicates in words[]


#Removing Duplicate Classes
classes = sorted(list(set(classes)))


#Printing length of lists we formed
print(len(documents),"Documents \n")
print(len(classes),"Classes \n")
print(len(words), "Stemmed Words ")


training = []
output = []


#Creating empty array for output
output_empty = [0] * len(classes)

```

```

#Creating Training set and bag of words for each sentence
for doc in documents:
    bag = [] #Initializing empty bag of words

    pattern_words = doc[0] #Storing list of tokenized words for the documents[] to pattern_words
    #print(pattern_words)

    #Again Stemming each word from pattern_words
    pattern_words = [stemmer.stem(word.lower()) for word in pattern_words]
    #print(pattern_words)

    #Creating bag of words array
    for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)

    #It will give output 1 for current tag and 0 for all other tags
    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1
    training.append([bag, output_row])
    random.shuffle(training) #Shuffling the data or features
    training = np.array(training) #Converting training data into numpy array

#Creating Training Lists
train_x = list(training[:,0])
train_y = list(training[:,1])
tf.reset_default_graph() #Reset Underlying Graph data

#Building our own Neural Network
net = tflearn.input_data(shape=[None, len(train_x[0])])
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, len(train_y[0]), activation="softmax")

```

```

net = tflearn.regression(net)

#Defining Model and setting up tensorboard
model = tflearn.DNN(net, tensorboard_dir="tflearn_logs")

#Now we have setup model, now we need to train that model by fitting data into it by model.fit()
model.fit(train_x, train_y, n_epoch=1000, batch_size=8, show_metric=True)
model.save("model.tflearn")

import pickle

#Dumping training data by using dump() and writing it into training_data in binary mode
pickle.dump({"words":words, "classes":classes, "train_x":train_x, "train_y":train_y},
open("training_data", "wb"))

#Restoring all data structure
data = pickle.load(open("training_data","rb"))
words = data['words']
classes = data['classes']
train_x = data['train_x']
train_y = data['train_y']

with open("intents.json") as json_data:
intents = json.load(json_data)

model.load("./model.tflearn")

def clean_up_sentence(sentence):

# Tokenizing the pattern
sentence_words = nltk.word_tokenize(sentence) #Again tokenizing the sentence

#Stemming each word from the user's input
sentence_words= [stemmer.stem(word.lower()) for word in sentence_words]

return sentence_words

```

#Returning bag of words array: 0 or 1 or each word in the bag that exists in as we have declared in above lines

```
def bow(sentence, words, show_details=False):
```

```
#Tokenizing the user input
```

```
sentence_words = clean_up_sentence(sentence)
```

```
#Generating bag of words from the sentence that user entered
```

```
bag = [0]*len(words)
```

```
for s in sentence_words:
```

```
for i,w in enumerate(words):
```

```
if w == s:
```

```
bag[i] = 1
```

```
if show_details:
```

```
print("Found in bag: %s"% w)
```

```
return(np.array(bag))
```

```
context = {} #Create a dictionary to hold user's context
```

```
ERROR_THRESHOLD = 0.25
```

```
def classify(sentence):
```

```
#Generating probabilities from the model
```

```
results = model.predict([bow(sentence, words)])[0]
```

```
#Filter out predictions below a threshold
```

```
results = [[i,r] for i,r in enumerate(results) if r>ERROR_THRESHOLD]
```

```
#Sorting by strength of probability
```

```
results.sort(key=lambda x: x[1], reverse=True)
```

```
return_list = []
```

```
for r in results:
```

```
return_list.append((classes[r[0]], r[1]))
```



```

# return tuple of intent and probability
return return_list

def response(sentence, userID='123', show_details=False):
    results = classify(sentence)

    #If we have a classification then find the matching intent tag
    if results:

        #Loop as long as there are matches to process
        while results:
            for i in intents['intents']:

                #Find a tag matching the first result
                if i['tag'] == results[0][0]:

                    #Set context for this intent if necessary
                    if 'context_set' in i:
                        if show_details: print ('context:', i['context_set'])
                        context[userID] = i['context_set']

                    # check if this intent is contextual and applies to this user's conversation
                    if not 'context_filter' in i or \
                        (userID in context and 'context_filter' in i and i['context_filter'] == context[userID]):
                        if show_details: print ('tag:', i['tag'])

                    #A random response from the intent
                    return print(random.choice(i['responses']))

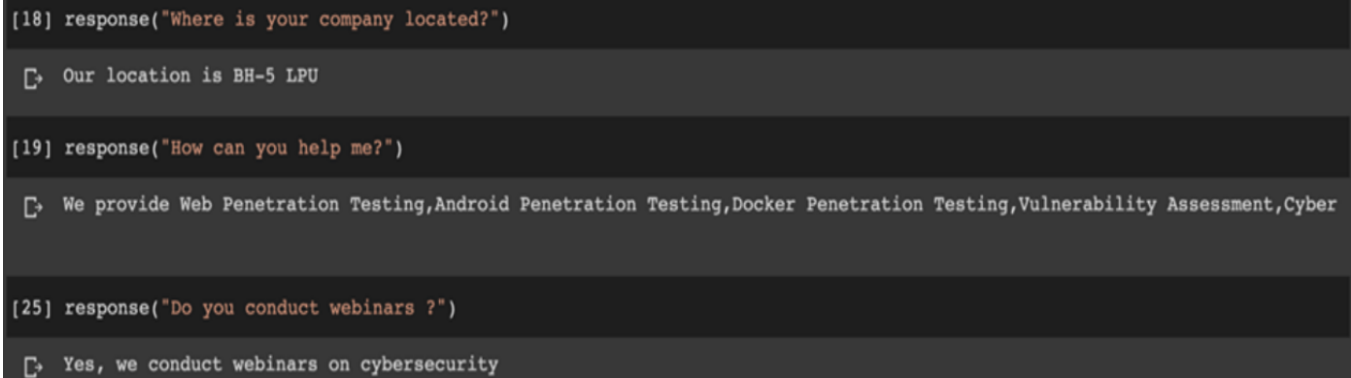
            results.pop(0)

```

CHAPTER 6

SCREENSHOTS AND RESULTS

Successful Execution Screenshot



```
[18] response("Where is your company located?")  
[19] response("How can you help me?")  
[25] response("Do you conduct webinars ?")
```

The screenshot displays a chatbot interface with three prompts and their corresponding responses. The prompts are: "Where is your company located?", "How can you help me?", and "Do you conduct webinars ?". The responses are: "Our location is BH-5 LPU", "We provide Web Penetration Testing,Android Penetration Testing,Docker Penetration Testing,Vulnerability Assessment,Cyber", and "Yes, we conduct webinars on cybersecurity".

Github Link

<https://github.com/Arushi247/CyberArmour/tree/main>

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, the AI chatbot in the field of cybersecurity offers a valuable solution for organizations looking to enhance their cybersecurity measures. By providing real-time responses to user queries, monitoring network activities, and alerting security personnel to potential threats, the chatbot can help to improve the efficiency and effectiveness of cybersecurity operations. The chatbot's multi-layered approach, combined with its ability to communicate in multiple languages, ensures that it can cater to a wide range of users and provide a personalized and efficient user experience. Overall, the AI chatbot represents an innovative and effective approach to cybersecurity that can help organizations to protect their digital assets against emerging threats.

Here are some potential future enhancements for an AI chatbot designed for cybersecurity:

1. **Multilingual support:** In today's global business environment, many organizations operate in multiple countries and deal with customers and partners who speak different languages. Enhancing an AI chatbot's natural language processing capabilities to support multiple languages can increase its effectiveness and usefulness for organizations operating in diverse markets.
2. **Integration with other security technologies:** AI chatbots can be integrated with other security technologies such as intrusion detection systems (IDS), firewalls, and security information and event management (SIEM) tools to enhance their ability to detect and respond to potential threats.
3. **Advanced analytics:** AI chatbots can be enhanced with advanced analytics capabilities such as machine learning algorithms to improve their ability to detect patterns and anomalies in data, and identify potential threats before they occur.
4. **Improved authentication:** AI chatbots can be integrated with advanced authentication technologies such as biometrics or multi-factor authentication to provide enhanced security for user accounts and prevent unauthorized access.
5. **Cloud-based deployment:** Cloud-based deployment can enable organizations to easily deploy and scale AI chatbots for cybersecurity purposes, without the need for significant infrastructure investments.
6. **Enhanced privacy and data protection:** AI chatbots can be designed to incorporate advanced privacy and data protection features such as end-to-end encryption and anonymization techniques to ensure the protection of sensitive information.
7. **Predictive capabilities:** AI chatbots can be enhanced with predictive capabilities that can help

organizations anticipate and prevent potential cybersecurity threats before they occur.

Overall, continuous development and enhancement of AI chatbots can lead to increased effectiveness and efficiency in cybersecurity operations, making them a valuable asset to organizations looking to safeguard their digital assets from potential cyber threats.

REFERENCES

❖ <https://www.edureka.co/blog/how-to-make-a-chatbot-in-python/>

❖

<https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077>

❖ <https://www.youtube.com/channel/UCv6Uw36LRbYnX4HDxKPguKg>