Article

Not peer-reviewed version

AICyberSec: A Cyber Security Tool

Dawid Pasamonik , Bartłomiej Smolarski , Weronika Strynkowska , Patrycja Lubicka , Marlena Eksman , Piotr Perliński , Jarosław Krzywanski , Yuriy Povstenko , Marcin Sosnowski , Ireneusz Szcześniak , Bożena Woźna-Szcześniak *

Posted Date: 6 November 2023

doi: 10.20944/preprints202311.0293.v1

Keywords: social media; artificial intelligence; machine learning; chatbots; cybersecurity; emerging applications; coding techniques

Article

AICyberSec: A Cyber Security Tool

Dawid Pasamonik [1], Bartłomiej Smolarski [1], Weronika Strynkowska [1], Patrycja Lubicka [1], Marlena Eksman [1], Piotr Perliński [1], Jarosław Krzywanski [1], Yuriy Povstenko [1], Marcin Sosnowski [1], Ireneusz Szcześniak [2] and Bożena Woźna-Szcześniak [1,*]

[1] Faculty of Science and Technology, Jan Dlugosz University in Czestochowa, Armii Krajowej 13/15, 42–200 Czestochowa, Poland; pasamonikdawid@interia.pl (D.P.); smolarski123@gmail.com (B.S.); w.strynkowska@interia.pl (W.S.); plubicka@gmail.com (P.L.); marlenaeksman4@o2.pl (M.E.); piotrperlinski1@gmail.com (P.P.); j.krzywanski@ujd.edu.pl (J.K.); j.povstenko@ujd.edu.pl (Y.P.); m.sosnowski@ujd.edu.pl (M.S.); b.wozna@ujd.edu.pl (B.W.S.)

[2] Department of Computer Science of the Częstochowa University of Technology, Dąbrowskiego 73, 42-200 Częstochowa, Poland; ireneusz.szczesniak@pcz.pl

* Correspondence: b.wozna@ujd.edu.pl

Abstract: Chatbots are increasingly used as tools for disseminating knowledge and information across various sectors of business, without the need for human intervention. With the continuous advancement of technology and the internet, the field of cybersecurity is also expanding, leading to the development of newer and more sophisticated systems to counter emerging threats. However, a fundamental question arises: how can one determine what to look for, where to acquire the necessary knowledge, and how to initiate this process? Recognizing the need for more effective tools in the market to educate individuals about online threats and self-protection, this paper aims to bridge the gap in the existing literature. The Artificial Intelligence-based Cybersecurity Chatbot (AICyberSec) has been developed to provide a versatile cybersecurity knowledge resource. This initiative is in response to the absence of a comprehensive solution in the market that effectively informs individuals about online threats and offers guidance on how to safeguard themselves. Furthermore, this chatbot serves as a valuable application for identifying potential hazards in cyberspace and can also function as an educational tool to facilitate the learning process.

Keywords: social media; artificial intelligence; machine learning; chatbots; cybersecurity; emerging applications; coding techniques

1. Introduction

With the beginning of the 21st century, computers and all internet-enabled devices gained significant importance. Computers, various mobile devices and a multitude of applications became ubiquitous in the daily routines of countless individuals, gradually becoming indispensable components of our existence. These technologies began to serve many purposes, spanning entertainment, work, and business, among others. However, the main reason for owning of these internet-enabled devices is the ability to quickly access information from around the world. This capability significantly impacts the personal development of each user and facilitates the fast and easy exchange of information [1].

Unfortunately, using this equipment brings many risks and associated consequences that the average user of computing devices may need to be aware of. The rapid development of technology means that newer and newer threats are emerging constantly, posing a significant danger to all users. The constant evolution of threats means that combating all hazards must relentlessly employ the latest defence methods [2]. Cybersecurity knowledge is also expanding with the development of technology and the internet, and leading to the creation of newer and more advanced systems to protect against these threats.

However, the question arises of how to know what to look for, where to get the proper knowledge on the subject, and where to start. It is well known that many global and domestic corporations

 doi:10.20944/preprints202311.0293.v1

use chatbots to improve their contact with customers and make it easier for them to access specific information. Thus, a chatbot is a computer program that employs artificial intelligence (AI) to simulate conversations with a human, usually on a particular range of topics. It can carry on a conversation with the user and fully understand them. Conversation is the primary function of a chatbot operating with the assistance of an adequately introduced knowledge base and algorithms [3].

Chatbots communicate directly with the user, allowing them to access their desired Furthermore, many companies use chatbots to inform potential customers and users of their websites or applications. Chatbots are also used in the financial industry (mainly banking), tourism, retail, telecommunications, and media industries. They are becoming tools for disseminating knowledge and information in many branches of business, which does not require the intervention of a second human. Thus, they are increasingly important for knowledge transfer and acquisition [4].

Leading companies have introduced chatbots into their communities, including Facebook, Google, and Apple. The same trend is evident in business, administration, and medicine. Chatbots play a significant role in computer games and "casual" conversations in the entertainment sector. They also find applications in education, where chatbots serve as tools to assist in learning [3].

Many types of chatbots are used and discussed in the literature. The most significant number of bots are still applied in the social media area, but they can have different intentions. Here we can distinguish between bots with benign, neutral or malicious intentions [5]. The following examples of chatbots with benign intentions can be listed in today's social media.

BBC Mundo for Facebook Messenger is a product of News Labs. The bot sends subscribers a list of news headlines each day that link back to the Mundo website. Users can request more headlines or share stories from within Messenger. Editors can also monitor metrics to determine how much traffic the bot directs back to the BBC website, which can then be used to inform further work on the platform. Twitter bots automatically created and tweeted graphics showing voting results on the nights of the EU Referendum and U.S. presidential elections. The EU Referendum bot was built after receiving a request from the Visual Journalism team and extended the code so it could be reused during the U.S. election. The authors hope the code behind these bots will become a standard part of the BBC's election coverage.

Launching initially with BBCUzbekBot, the Uzbek Telegram bots can deliver news where audiences can't access the BBC's website. In addition to the bots themselves, a web-based tool was created that allows BBC editors to add custom commands and features without modifying code.

News Labbers Lei He, Alex Norton, and Rachel Wilson helped develop the winning entry at the IBM Watson Build a Bot hackathon. Since news is one of several sources used to power interaction IBM Watson wingBot employs a combination of APIs, experimenting with text-to-speech and applying artificial intelligence techniques. The author intends to explore it further in bot-building for BBC News [6].

Other interesting groups are chatbots with neutral intentions. More and more of the witty and thought-provoking content on Twitter is generated by bots, artificial systems that write their material and vie for our attention just as humans do. Real people follow artificial bots for subtle and diverse reasons, but a significant motive is undoubtedly Twitter. Twitter is an intelligent environment for automated wit, with its inventive Twitterbot named MetaphorMagnet [7].

In contrast to the other bots, the ErtBot chatbot was developed to detect and construct accounts of technological practices as instances of energy consumption. As such, it formulates suggestions on how to save energy and take environmental actions [8].

Finally, File-sharing Bots are chatbots with malicious intentions. These bots take the user's query term (e.g., a popular movie or artist's album) and respond to the query stating that they have the file available for download, providing a link. The unexpecting user clicks on the link, downloads and opens it, and unknowingly has infected their computer.

Other examples of chatbots with malicious intentions are spam bots and zombie bots. These bots flood the inbox with spam and interrupt chats by sending unsolicited instant messages. Some

 doi:10.20944/preprints202311.0293.v1

aggressive advertisers use these bots to target individuals based on demographic information obtained from the user's profile. These bots are usually easy to spot because they typically send you a link to click on along with some enticing comment to get you to click.

On the other hand, the term "zombie bots" refers to computers that have been compromised, along with hundreds or thousands of different computers, as part of a botnet. These computers coordinate large-scale attacks where all the zombie computers act in unison, carrying out commands sent by the botnet owner. These bots are more challenging to detect and are more stealthy. Quite often, victims of this type of infection don't even know that their systems are infected [9–12].

All the bots presented above are examples of the ones occurring nowadays, and new types of bots are constantly being created.

The literature review reveals that there is a need for a tool that effectively informs people about online threats and how to protect themselves. The present paper addresses this gap by introducing the developed AICyberSec chatbot, which offers a cybersecurity dictionary with a wide-ranging knowledge base. This chatbot was created in response to the absence of a tool in the market that effectively educates people about online threats and self-protection. Consequently, the chatbot empowers users to enhance their cybersecurity knowledge.

The main functions of the newly developed AICyberSec chatbot are:

- Clarification of concepts and nomenclature in the field of cyber security;
- Expanding one's knowledge in the field of cybersecurity;
- Enhancing one's cyber security by being able to obtain information about the threats present on the network;
- Providing the ability to gain knowledge about securing one's data and devices in cyberspace.

2. An Object and Methods

The developed AICyberSec bot employs scripts that construct the entire framework upon which the chatbot operates. These scripts include mechanisms to ensure it avoids self-contradiction. Thanks to machine learning (ML) procedures, the chatbot is capable of autonomous learning. With each program run, the chatbot continues to learn and evolves to provide increasingly accurate responses to user queries. Through numerous training epochs, the chatbot progressively enhances its ability to analyze text input.

The library contains essential topics related to forensics and cybercrime, consisting of 150 terms. Whenever a user inquires about topics related to cybercrime, the chatbot responds with a definition or brief description of the content. Thanks to its extensive library containing various questions, the chatbot can identify meaningful words in the user's query. Furthermore, the chatbot can engage in basic conversations as it possesses a rudimentary self-awareness, such as awareness of the current time and its mood. The database was developed using the cybersecurity dictionary authored by Tomasz Pączkowski [13].

3. Results and Discussion

After launching the application, the chatbot remains in standby mode, waiting for the user to initiate a conversation to avoid interrupting the user's question formulation process. The chatbot is built using natural language processing (NLP) and neural networks (ANN) in Python. The database is stored in a JSON file format, a lightweight computer data exchange format based on a subset of JavaScript. Although the program is written in Python, the decision was made to utilize JavaScript's JSON format due to its simplicity and efficiency. The library encompasses various components, including multiple greetings, farewells, and custom questions.

When a user asks standard questions, the chatbot learns specific sentence structures and words so that it can smoothly, flexibly choose the correct answer or comments to make the conversation resemble a human discussion as much as possible instead of throwing out a text saying, "doesn't understand the content", "can you ask another question". This creates a kind of continuous learning

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 November 2023 doi:10.20944/preprints202311.0293.v1

for the bot, which tries to adapt to the user as best as possible. The database-building process starts by entering "intents," or intents by which chatbot communication occurs. Each of these consists of three subcategories, which include:

- Tag - a keyword that categorizes specific data.
- Patterns - this section contains the questions that the user asks.
- Responses - here are the chatbot's answers to our questions.

After creating the library, whether it contains one or a hundred items, a new file named "Training.py" is generated. This file encompasses the entire script responsible for training a deep-learning model tasked with classifying and interpreting user inquiries directed at the chatbot. The procedure begins with the importing of essential modules, including random, json, numpy, and pickle. Next, the NumPy library is installed via the command line using the 'pip' package manager, along with the 'nltk' (Natural Language Toolkit) module for natural language processing and machine learning components. This configuration is applied during each program execution.

To enhance the chatbot's database search and response accuracy, the implementation of 'nltk.stem' into WordNet Lemmatizer is recommended to handle word variations (see Listing 1).

Listing 1: The screenshot of the chatbot's code with the import nltk libraries

```
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from keras.optimizers import SGD
import random

import nltk
from nltk.stem import WordNetLemmatizer
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')

lemmatizer = WordNetLemmatizer()
import json
import pickle

words = []
classes = []
documents = []
ignore_letters = ['!','?',',','.']
```

After implementing the aforementioned crucial components into the system, which do not interfere with the rest of the chatbot's code, the process of loading the database file containing content to be pass to the user is initialized (see Listing 2).

After completing the script, we can test our current code by using the 'print' command to display the content we have already included. This enables us to visualize all the comments we have integrated. It's important to note that when we intend to modify the library containing our text, we will need to debug or reimplement the base due to the inevitable changes in a separate file. Additionally, the script with 'words = sorted(set(words))' is noteworthy as it helps prevent duplicate comments from affecting the code's functionality.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 November 2023 doi:10.20944/preprints202311.0293.v1

Listing 2: The screenshot of the Database

```
"intents":[
 {
   "tag": "greeting",
   "patterns": ["Hi", "How are you", "Is anyone there?",
                "Hello", "Good day"],
   "responses":["Hello, thanks for visiting",
                "Good to see you again",
                "Hi there, how can I help?"],
   "context_set": ""
 },
 {
   "tag": "goodbye",
   "patterns": ["Bye", "See you later", "Goodbye", "Elo"],
   "responses": ["See you later, thanks for visiting",
                 "Have a nice day",
                 "Bye! Come back again soon.", "Elo"]
 },
 {
   "tag": "thanks",
   "patterns": ["Thanks", "Thank you", "That's helpful" ],
   "responses": ["Happy to help!", "Any time!", "My pleasure"]
 },
 {
   "tag": "hours",
   "patterns": ["what hours are you open?",
                "what are your hours?", "when are you open?",
                "when is the time to contact ?",
  "responses": ["We're open every day from 9AM to 9PM",
                "Our working hours are 9AM to 9PM every day"]
 } ...
```

Training is a critical phase in the development of the chatbot, as it requires the incorporation of appropriate scripts to educate our bot [14–16]. Consequently, the training step is a machine learning process that remains a key step of our program each time we execute the program. It is advisable to establish a loop with input documents, output documents, words, databases, and lists that operates seamlessly without user intervention, facilitating the fine-tuning of the neural network during training (see Listing 3).

Listing 3: The idea of using the loop with input documents, output documents, words, bases and lists

```
for intent in intents['intents']:
    for pattern in intent['patterns']:
        #tokenize each word
        word = nltk.word_tokenize(pattern) words.extend(word)
        #add documents in the corpus
        documents.append((word, intent['tag']))
        # add to our classes list
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
print(documents)
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 November 2023 doi:10.20944/preprints202311.0293.v1

Subsequently, the neural network is partitioned into two sequential models that function concurrently in a layered manner. This entails the division of our current neural network into the initial sequence, which operates, and the second sequence, comprising layers responsible for the ongoing training of our network. This can be likened to the functioning principle of the human brain, which simultaneously learns, analyzes, and reprocesses acquired information at any given moment. Our artificial neural network consists of three dense layers, including two hidden layers with 128 and 64 rectified linear units, respectively, along with a softmax-type output layer (see Listing 4).

Listing 4: The screenshot with the model definition

```
# Create model - 3 layers. First layer 128 neurons,
# second layer 64 neurons and 3rd output layer contains
# number of neurons equal to number of intents
#to predict output intent with softmax
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),),
          activation='relu'))
model.add(Dropout (0.5))
model.add(Dense(64, activation-'relu'))
model.add(Dropout (0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))

# Compile model. Stochastic gradient descent with Nesterov
# accelerated gradient gives good results for this model
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd,
              metrics=['accuracy'])

#fitting and saving the model
hist = model.fit(np.array(train_x), np.array(train_y),
                 epochs=200, batch_size=5, verbose=1)
model.save('chatbot_model.h5', hist)

print("model created")
```

In the information theory sense, there is a connection between categorical cross-entropy and entropy, mainly because categorical cross-entropy is used as a loss function in machine learning. This connection between entropy in information theory and categorical cross-entropy lies in their mathematical form. Categorical cross-entropy measures the "surprise" or "information content" associated with the predicted probabilities compared to the actual possibilities. When the predicted probabilities align perfectly with the actual chances, the loss is minimized (approaching zero), indicating no surprise or uncertainty. Conversely, when there is a mismatch between the two distributions, the loss increases, implying a higher level of surprise or uncertainty. In other words, categorical cross-entropy in machine learning quantifies the information content or "surprise" associated with the model's predictions compared to the ground truth, drawing a connection to the concept of entropy in information theory. Minimizing categorical cross-entropy during training helps the model learn to make predictions as close as possible to the actual data distribution.

Our model uses categorical cross-entropy as the loss function and the stochastic gradient descent method as the optimizer. In the penultimate step of our script, we put all the information with the method and speed of learning that our program will use to learn as efficiently as possible from the information it acquires. The learning speed must be well balanced against the bot's content because if it has lots of data to process and short training periods may cause an error to arise, it will start

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 November 2023 doi:10.20944/preprints202311.0293.v1

confusing its scripts and deteriorate the chatbot's performance. A comment „model created" is printed at the end of the process to inform that the chatbot model has been created.

The final step is the graphical user interface (GUI) building procedure (Listing 5). The visual environment allows for the Chatbot's graphical presentation in the operating system and the interface for communicating with the trained chatbot. The GUI uses the Tkinter module, which is responsible for building the structure of the desktop application and then capturing the user's messages. The model predicts the user's message tag and selects the response from the intents.JSON file.

The graphical structure begins by specifying the program's dialogue box size. In our case, it is 800 x 1000 px. Next is the creation of the chat window, which will display the text of the conversation. A scrollbar was also created for the GUI so the user could return to earlier issues raised in the conversation. The next step was to create an action button for the user to send messages. The button colours were written in a hexagonal format. Next, a dialogue box was designed that allowed the user to type in the content of a potential ques-tion. The final step was placing the abovementioned elements in the chat window in spe-cific locations on the X and Y axes. Adding a loop responsible for the continuity of the process is also necessary.

Listing 5: The chatbot's code for the graphical user interface (GUI)

```
root = Tk()
root.title("Chatbot")
root.geometry("800x1000")
root.resizable (width=FALSE, height=FALSE)
#Create Chat window
ChatBox = Text(root, bd=0, bg="white", height="8",
          width="50", font="Arial",)
ChatBox.config(state=DISABLED)

#Bind scrollbar to Chat window
scrollbar = Scrollbar (root, command-ChatBox.yview,
                       cursor="heart")
ChatBox['yscrollcommand'] = scrollbar.set

#Create Button to send message
SendButton = Button(root, font=("Verdana",12,'bold'),
             text="Send", width="16", height=7,
             bd=0, bg="#f9a602", activebackground="#3c9d9b",
             fg='#000000', command send)

#Create the box to enter message
EntryBox = Text(root, bd=0, bg="white",width="29",
                height="7", font="Arial")
#EntryBox.bind("<Return>", send)

#Place all components on the screen
scrollbar.place(x=752,y=12, height=772)
ChatBox.place(x=12,y=12, height=772, width=740)
EntryBox.place(x=256, y=802, height=180, width=530)
SendButton.place(x=12, y=802, height=90)

root.mainloop()
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 November 2023 doi:10.20944/preprints202311.0293.v1

The developed AICyberSec bot is a selfcontained application uploaded from an external source and independently working individual computer-type media. The tool aligns with recent advances in education [17,18]. The tool was created using Python 3 programming language, and the database was created in JSON format. AI CyberSec bot is an application with cyber security knowledge based on an artificial intelligence approach.

To use the application, the user asks a question of concern, and the chatbot will respond by explaining the concept. A sample dialogue window during a conversation with the AICyberSec chatbot is shown in Figure 1.

The AICyberSec chatbot correctly identifies and provides the contents of the database as a response to the provided user query.

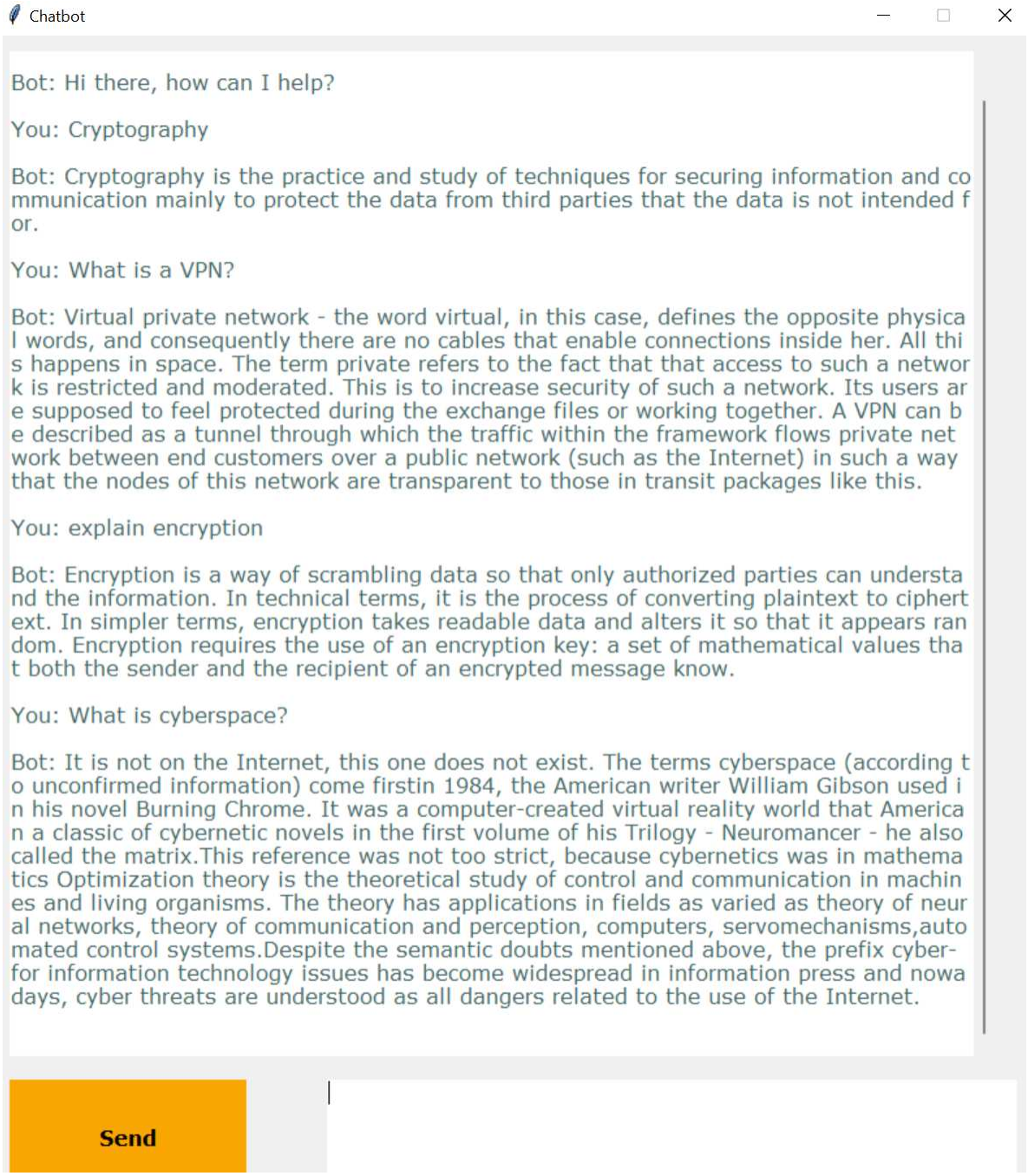

Figure 1. A sample dialogue window of the AICyberSec chatbot.

 doi:10.20944/preprints202311.0293.v1

4. Conclusions

This project aims to underscore the significance of artificial intelligence, specifically chatbots, in modern education. It's noteworthy that in the future, artificial intelligence may potentially replace traditional textbooks, contributing to a more environmentally friendly approach to learning. The developed AICyberSec chatbot introduces an innovative approach to scientific literature, particularly in the domain of cybersecurity. Cybersecurity awareness is a critical factor impacting people's quality of life, and public awareness on this issue still needs significant growth. It is imperative for humanity to comprehend the dangers of the internet, and the developed tool plays a vital role in elucidating the hazards that exist in cyberspace.

The paper presents an illustrative use case of a chatbot functioning as a cybersecurity dictionary. Furthermore, its extensive knowledge base offers versatility, making it feasible to adapt and create new chatbots for different domains.

The chatbot developed in this paper operates through an external application, requiring no internet access, and can be employed offline at any time and location. Users gain immediate access to the application and the requisite libraries to utilize the chatbot. Its operational mechanism is based on the following principles:

- Machine learning - a concept within artificial intelligence in which we let the ma-chine learn things on its own if it is equipped with enough data and computing power;
- NLTK (Natural Language Toolkit) – the library that uses correlations, creates a unique "brain" of the program, which quickly learns independently;

The presented application works better than a text document because it is faster, does not make factual errors, and all generated answers are correct.

Author Contributions: Conceptualization: D.P., B.S., W.S., P.L., M.E., P.P. and J.K.; methodology: D.P., B.S., W.S., P.L., M.E., P.P., J.K.; software: D.P., B.S., W.S., P.L., M.E., P.P., J.K. and I.S.; validation: D.P., B.S., W.S., P.L., M.E., P.P., J.K., I.S. and B.W.S.; formal analysis: D.P., B.S., W.S., P.L., M.E., P.P., J.K. and B.W.S.; investigation: D.P., B.S., W.S., P.L., M.E., P.P. and Y.P.; resources: D.P., B.S., W.S., P.L., M.E., P.P. and J.K.; data curation: D.P., B.S., W.S., P.L., M.E. and P.P.; writing-original draft preparation: D.P., B.S., W.S., P.L., M.E., P.P., J.K., B.W.S and Y.P.; writing-review and editing: J.K., Y.P., I.S. and B.W.S.; visualization: D.P., B.S., W.S., P.L., M.E., P.P. and I.S.; supervision: J.K., M.S., Y.P., I.S. and B.W.S.; project administration I.S and B.W.S.; funding acquisition Y.P., J.K., M.S., I.S. and B.W.S.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Unavailable due to privacy restrictions.

Acknowledgments: The authors thank Dr. Tomasz Pączkowski for permission to use his publication to create a database for the chatbot.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lally, E. *At Home with Computers*; Routledge. Taylor & Francis Group, 2020.
2. Conti, M.; Dargahi, T.; Dehghantanha, A. Cyber Threat Intelligence: Challenges and Opportunities. In Proceedings of the Cyber Threat Intelligence; Dehghantanha, A.; Conti, M.; Dargahi, T., Eds. Springer,Cham, 2018, Vol. 70, *Advances in Information Security*, pp. 1–6. https://doi.org/10.1007/978-3-319-73951-9_1.
3. Ahmad, N.A.; Hamid, M.H.C.; Zainal, A.; Rauf, M.F.A.; Adnan, Z. Review of Chatbots Design Techniques. *International Journal of Computer Applications* **2018**, *181*, 7–10. https://doi.org/10.5120/ijca2018917606.
4. Schneider, A.; Janowska, A. Chatboty w Polsce 2020. Wyniki badania ilościowego oraz benchmark polskich chatbotów w finansach, telekomunikacji i mediach, turystyce oraz retailu. on-line, https://symetria.pl/chatboty-w-polsce/Chatboty-w-Polsce-2020.pdf, 2020. in Polish.

5. Orabi, M.; Mouheb, D.; Al Aghbari, Z.; Kamel, I. Detection of Bots in Social Media: A Systematic Review. *Information Processing & Management* **2020**, *57*, 102250. https://doi.org/https://doi.org/10.1016/j.ipm.2020.102250.
6. Heinrich, G.; Oduyemi, R. Bots. on-line, https://bbcnewslabs.co.uk/projects/bots, 2022.
7. Veale, T.; Valitutti, A.; Li, G. Twitter: The Best of Bot Worlds for Automated Wit. In Proceedings of the Distributed, Ambient, and Pervasive Interactions. DAPI 2015; Streitz, N.; Markopoulos, P., Eds. Springer, Cham, 2015, Vol. 9189, *Lecture Notes in Computer Science*, pp. 689–699.
8. Wilkie, A.; Michael, M.; Plummer-Fernandez, M. Speculative Method and Twitter: Bots, Energy and Three Conceptual Characters. *The Sociological Review* **2015**, *63*, 79–101. https://doi.org/10.1111/1467-954X.12168.
9. Pacinelli, R. Blog: The best media. on-line, https://thebestmedia.com/malicious-bots-what-are-they-and-how-can-you-avoid-them, 2023.
10. Carrillo-Mondéjar, J.; Martinez, J.; Suarez-Tangil, G. On how VoIP attacks foster the malicious call ecosystem. *Computers & Security* **2022**, *119*, 102758. https://doi.org/https://doi.org/10.1016/j.cose.2022.102758.
11. Kumar, Y.; Subba, B. Stacking Ensemble-Based HIDS Framework for Detecting Anomalous System Processes in Windows Based Operating Systems Using Multiple Word Embedding. *Computers and Security* **2023**, *125*. https://doi.org/10.1016/j.cose.2022.102961.
12. Azad, M.A.; Morla, R.; Salah, K. Systems and methods for SPIT detection in VoIP: Survey and future directions. *Computers & Security* **2018**, *77*, 1–20. https://doi.org/https://doi.org/10.1016/j.cose.2018.03.005.
13. Pączkowski, T. *Dictionary of Cyber Security*; Szkoła Policji w Katowicach, 2017. (in Polish).
14. Skrobek, D.; Krzywanski, J.; Sosnowski, M.; Kulakowska, A.; Zylka, A.; Grabowska, K.; Ciesielska, K.; Nowak, W. Implementation of deep learning methods in prediction of adsorption processes. *Advances in Engineering Software* **2022**, *173*, 103190. https://doi.org/https://doi.org/10.1016/j.advengsoft.2022.103190.
15. Krzywanski, J.; Blaszczuk, A.; Czakiert, T.; Rajczyk, R.; Nowak, W. Artificial intelligence treatment of NOx emissions from CFBC in air and oxy-fuel conditions. In Proceedings of the Proceedings of the 11th International Conference on Fluidized Bed Technology, Beijing, China, 2014; p. 619–624.
16. Krzywanski, J.; Sztekler, K.; Bugaj, M.; Kalawa, W.; Grabowska, K.; Chaja, P.R.; Sosnowski, M.; Nowak, W.; Mika, L.; Bykuć, S. Adsorption chiller in a combined heating and cooling system: simulation and optimization by neural networks. *Bulletin of the Polish Academy of Sciences Technical Sciences* **2021**, *69*, e137054. https://doi.org/https://doi.org/10.24425/bpasts.2021.137054.
17. Guo, K.; Zhong, Y.; Li, D.; Chu, S.K.W. Effects of chatbot-assisted in-class debates on students' argumentation skills and task motivation. *Computers & Education* **2023**, *203*, 104862. https://doi.org/https://doi.org/10.1016/j.compedu.2023.104862.
18. Iku-Silan, A.; Hwang, G.J.; Chen, C.H. Decision-guided chatbots and cognitive styles in interdisciplinary learning. *Computers & Education* **2023**, *201*, 104812. https://doi.org/https://doi.org/10.1016/j.compedu.2023.104812.