**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **16-06-2020** | **Name:** | **Karthik J** |
| **Course:** | Cyber Security Go from Zero | **USN:** | **4AL16EC030** |
| **Topic:** |  | **Semester & Section:** | **8TH A** |
| **GitHub Repository:** | Karthik-J |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| Cyber Security Introduction "Cybersecurity is primarily about people, processes, and technologies working together to encompass the full range of threat reduction, vulnerability reduction, deterrence, international engagement, incident response, resiliency, and recovery policies and activities, including computer network operations, information assurance, law enforcement, etc."  Cybersecurity is the protection of Internet-connected systems, including hardware, software, and data from cyber attacks. It is made up of two words one is cyber and other is security. Cyber is related to the technology which contains systems, network and programs or data. Whereas security related to the protection which includes systems security, network security and application and information security.  It is the body of technologies, processes, and practices designed to protect networks, devices, programs, and data from attack, theft, damage, modification or unauthorized access. It may also be referred to as **information technology security**.  We can also define cybersecurity as the set of principles and practices designed to protect our computing resources and online information against threats. Due to the heavy dependency on computers in a modern industry that store and transmit an abundance of confidential and essential information about the people, cybersecurity is a critical function and needed insurance of many businesses. Why is cybersecurity important? We live in a digital era which understands that our private information is more vulnerable than ever before. We all live in a world which is networked together, from internet banking to government infrastructure, where data is stored on computers and other devices. A portion of that data can be sensitive information, whether that be intellectual property, financial data, personal information, or other types of data for which unauthorized access or exposure could have negative consequences.  Cyber-attack is now an international concern and has given many concerns that hacks and other security attacks could endanger the global economy. Organizations transmit sensitive data across networks and to other devices in the course of doing businesses, and cybersecurity describes to protect that information and the systems used to process or store it.  As the volume of cyber-attacks grows, companies and organizations, especially those that deal information related to national security, health, or financial records, need to take steps to protect their sensitive business and personal information. What Is Network Topology? Network topology refers to how various nodes, devices, and connections on your network are physically or logically arranged in relation to each other. Think of your network as a city, and the topology as the road map. Just as there are many ways to arrange and maintain a city—such as making sure the avenues and boulevards can facilitate passage between the parts of town getting the most traffic—there are several ways to arrange a network. Each has advantages and disadvantages and depending on the needs of your company, certain arrangements can give you a greater degree of connectivity and security.  There are two approaches to network topology: physical and logical. Physical network topology, as the name suggests, refers to the physical connections and interconnections between nodes and the network—the wires, cables, and so forth. Logical network topology is a little more abstract and strategic, referring to the conceptual understanding of how and why the network is arranged the way it is, and how data moves through it. Why Is Network Topology Important? The layout of your network is important for several reasons. Above all, it plays an essential role in how and how well your network functions. Choosing the right topology for your company’s operational model can increase performance while making it easier to locate faults, troubleshoot errors, and more effectively allocate resources across the network to ensure optimal network health. A streamlined and properly managed network topology can increase energy and data efficiency, which can in turn help to reduce operational and maintenance costs.  The design and structure of a network are usually shown and manipulated in a software-created network topology diagram. These diagrams are essential for a few reasons, but especially for how they can provide visual representations of both physical and logical layouts, allowing administrators to see the connections between devices when troubleshooting.  The way a network is arranged can make or break network functionality, connectivity, and protection from downtime. The question of, “What is network topology?” can be answered with an explanation of the two categories in the network topology.   1. **Physical** – The physical network topology refers to the actual connections (wires, cables, etc.) of how the network is arranged. Setup, maintenance, and provisioning tasks require insight into the physical network. 2. **Logical** – The logical network topology is a higher-level idea of how the network is set up, including which nodes connect to each other and in which ways, as well as how data is transmitted through the network. Logical network topology includes any virtual and cloud resources.   Effective network management and monitoring require a strong grasp of both the physical and logical topology of a network to ensure your network is efficient and healthy. Types of Network TopologyBUS TopologyRING TopologySTAR TopologyMESH TopologyTREE TopologyHYBRID TopologyNetwork Protocol A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network. Essentially, it allows connected devices to communicate with each other, regardless of any differences in their internal processes, structure or design. Network protocols are the reason you can easily communicate with people all over the world, and thus play a critical role in modern digital communications.  Similar to the way that speaking the same language simplifies communication between two people, network protocols make it possible for devices to interact with each other because of predetermined rules built into devices’ software and hardware. Neither [local area networks (LAN)](https://www.comptia.org/content/guides/what-is-a-local-area-network) nor [wide area networks (WAN)](https://www.comptia.org/content/guides/what-is-a-wide-area-network) could function the way they do today without the use of network protocols. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | | **16-06-2020** | **Name:** | **Karthik J** |  |
| **Course:** | | CNN for Computer Vision with Keras and TensorFlow in Python | **USN:** | **4AL16EC030** |  |
| **Topic:** | |  | **Semester & Section:** | **8th A** |  |
|  | **AFTERNOON SESSION DETAILS** | | | | |
|  | **My work:** <https://github.com/Karthikjsannakki/CNN>  **Image of session** | | | | |
|  | **Keras** [tf.keras](https://www.tensorflow.org/api_docs/python/tf/keras) is TensorFlow's high-level API for building and training deep learning models. It's used for fast prototyping, state-of-the-art research, and production, with three key advantages:   * User-friendly Keras has a simple, consistent interface optimized for common use cases. It provides clear and actionable feedback for user errors. * Modular and composable Keras models are made by connecting configurable building blocks together, with few restrictions. * Easy to extend Write custom building blocks to express new ideas for research. Create new layers, metrics, loss functions, and develop state-of-the-art models.  **Seaborn: Python's Statistical Data Visualization Library** One of the best but also more challenging ways to get your insights across is to visualize them: that way, you can more easily identify patterns, grasp difficult concepts or draw the attention to key elements. When you’re using Python for data science, you’ll most probably will have already used [Matplotlib](https://matplotlib.org/), a 2D plotting library that allows you to create publication-quality figures. Another complimentary package that is based on this data visualization library is [Seaborn](http://seaborn.pydata.org/), which provides a high-level interface to draw statistical graphics. NumPy NumPy is a Python package used for numerical computation. NumPy is one of the foundational packages for scientific computing with Python. NumPy's core data type is the array and NumPy functions operate on arrays. Installing NumPyInstalling NumPy Before NumPy's functions and methods can be used, NumPy must be installed. Depending on which distribution of Python you use, the installation method is slightly different. Install NumPy on Anaconda If you installed the Anaconda distribution of Python, NumPy comes pre-installed and no further installation steps are necessary.  If you use a version of Python from python.org or a version of Python that came with your operating system, the **Anaconda Prompt** and **conda** or **pip** can be used to install NumPy. Install NumPy with the Anaconda Prompt To install NumPy, open the **Anaconda Prompt** and type:  > conda install numpy  Type y for yes when prompted. Install NumPy with pip To install NumPy with **pip**, bring up a terminal window and type:  $ pip install numpy  This command installs NumPy in the current working Python environment. Verify NumPy installation To verify NumPy is installed, invoke NumPy's version using the Python REPL. Import NumPy and call the .\_\_version\_\_ attribute common to most Python packages.  In [1]:  import numpy as np  np.**version**  Out[1]:  '1.16.4'  A version number like '1.16.4' indicates a successful NumPy installation. Python Lists and NumPy Arrays NumPy is a Python package used for numerical calculations, working with arrays of homogeneous values, and scientific computing. This section introduces NumPy arrays then explains the difference between Python lists and NumPy arrays. Python Lists NumPy is used to construct homogeneous arrays and perform mathematical operations on arrays. A NumPy array is different from a Python list. The data types stored in a Python list can all be different.  python\_list = [ 1, -0.038, 'gear', True]  The Python list above contains four different data types: 1 is an integer, -0.038 is a float, 'gear' is a string, and 'True' is a boolean.  The code below prints the data type of each value store in python\_list.  In [1]:  python\_list = [1, -0.038, 'gear', True]  for item in python\_list:  print(type(item))  <class 'int'>  <class 'float'>  <class 'str'>  <class 'bool'>  Python.org downloads page showing download for Windows button | | | | |