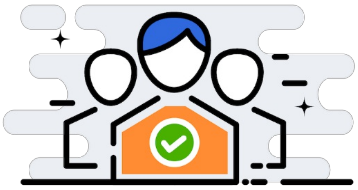
Table of Contents



Thanks for Choosing WISEN for your needs. Kindly refer your friends to WISEN.

Abbreviation

DNS	Domain Name Server
DoH	DNS over HTTPS
HTTP	Hyper Text Transfer Protocol
NTP	Network Time Protocol
UDP	User Datagram Protocol

Modified Title

Improving Robustness and Effective DNS Attack Detection using Neural Network

Chapter 1: Introduction

Abstract

Distributed Denial-of-Service (DDoS) attacks exhaust resources, leaving a server unavailable to legitimate clients. The Domain Name System (DNS) is a frequent target of DDoS attacks. Since DNS is a critical infrastructure service, protecting it from DoS is imperative. Many prior approaches have focused on specific filters or anti-spoofing techniques to protect generic services. DNS root nameservers are more challenging to protect, since they use fixed IP addresses, serve very diverse clients and requests, receive predominantly UDP traffic that can be spoofed, and must guarantee high quality of service.

This attack is made by leveraging the Domain Name System (DNS) technology through Domain Generation Algorithms (DGAs), a stealthy connection strategy that yet leaves suspicious data patterns. To detect such threats, advances in their analysis have been made. For the majority, they found Deep Learning (DL) as a solution, which can be highly effective in analyzing and classifying massive amounts of data. Although strongly performing, ML models have a certain degree of obscurity in their decision-making process. To cope with this problem, a branch of DL known as Explainable DL tries to break down the black-box nature of classifiers and make them interpretable and human-readable.

Our analysis reveals that the approach was successful in inferring significant DNS amplification DDoS activities including the recent prominent attack that targeted one of the largest anti-spam organizations. Moreover, the analysis disclosed the mechanism of such DNS amplification DDoS attacks.

General System

The Domain Name System (DNS) service is one of the core services in the internet functionality. Distributed Denial of Service (DDoS) attacks on DNS service typically consist of many queries coming from a large botnet. These queries are sent to the root name server or an authoritative name server along the domain chain. The targeted name server receives a high volume of requests, which may degrade its performance or disable it completely. Such attacks may also contain spoofed source addresses resulting in a reflection of the attack or may send requests that generate large responses (such as an ANY request) to use the DNS for amplification attacks. According to Akamais state of the internet report nearly 20% of DDoS attacks in Q1 of 2016 involved the DNS service. Moreover, even some of the Internets DNS root name servers were targeted.

One type of particularly hard to mitigate DDoS attacks are randomized attacks on the DNS service. In these attacks, queries for many different non-existent subdomains (subkeys) of the same primary domain (key) are issued). Since the result of a query to a new subdomain is not cached at the DNS resolver, these queries are propagated to the domain authoritative server, overloading both these servers and the open resolvers of the Internet Service Provider

Domain name system (DNS) is one of the most important technologies of the Internet. We can convert a domain name into an IP address using DNS. Without this service, the Internet would not be deployed as widely as it is now. DNS messages are normally built on top of UDP packets. Unlike in TCP, it is easy to forge the source address of UDP packets. As a result, DNS requests with a fake source address can easily be sent to a DNS server. In theory, any DNS server can answer any domain name resolution request; there are no protocol requirements that limit or filter request messages from client nodes. When DNS was invented, malicious activity utilizing DNS servers as packet reflectors was not extensive; however, as the Internet grew, attackers started to use this open operating policy to send traffic to victim nodes by forging DNS message source addresses. To prevent this activity, recent DNS servers have been configured to answer requests originating only from specific client nodes, typically filtered by source IP address. Unfortunately, there are more than a few improperly configured DNS servers in the wild; these are called open resolvers. The DNS protocol is still one of the major methods for attacking.

The Domain Name System (DNS) is a global, hierarchical, distributed database which serves, among other things, to map domain names to Internet Protocol (IP) addresses. While relatively straight forward in concept, in practice the global DNS is complex to the point of being arcane . For the purpose of this paper, we introduce a limited scope and vocabulary; the interested reader can find more depth in the operation and security of DNS in and . The domain name system operates as a query-response protocol, in which a query for a fully qualified domain name (FQDN) is made by a client, or endpoint, and is answered via an iterative process known as resolution. An FQDN is made up of a series of text labels separated by periods. For example, the FQDN www.google.com has three labels [www, google, com].

From a hierarchical perspective, the right-most label in an FQDN is the top-level-domain (TLD) and the each subsequent label represents a subdomain of the FQDN created from all the prior labels. For instance www.google.com is a subdomain of google.com, which in turn is a subdomain of com. The term domain is often used to refer to both an FQDN and the scope of its possible subdomains. Thus www.google.com, mail.google.com, inbox.google.com, and photo.google.com are all subdomains of the domain google.com. The TLD is a single label and is considered public in the sense that its subdomains are available for registration and not controlled by the TLD. The TLDs are limited in number and controlled by the Internet Corporation for Assigned Names and Numbers (ICANN). In some cases, subdomains of a TLD are also managed as public domains, most notably domains like co.uk, and are considered public suffixes and create extended TLDs (eTLD).2 A second-level-domain (SLD) is privately owned and the direct subdomain of a public suffix, or eTLD. Examples of second-level-domains include google.com and google.co.uk. The SLD is sometimes referred to as the base domain.

Within the Domain Name System, authoritative name servers are servers which hold authoritative, or definitive, answers for a certain portion of the database, generally a specific domain. If these authoritative servers are not functioning properly, Internet traffic to the domains for which they are authoritative may be interrupted or completely disrupted. For this reason, companies often have multiple authoritative name servers for their domains.

While it is possible for an endpoint to resolve DNS queries themselves, in practice, most devices rely on large recursive resolvers to perform resolution on their behalf. Internet Service Providers provide recursive resolvers for their customers, for example. Many recursive resolvers are configured to answer queries only for devices in their network, thereby limiting the resource demands on those network appliances. There are public resolvers, such as those operated by Google, openDNS, and Cloudflare, designed to handle recursion for any endpoint selecting their service.3 On the other hand, there are also a large number of devices on the Internet that, through misconfiguration or otherwise, will act as recursive resolvers for any client but are not announced as public resolvers. These are known as open resolvers and are frequently leveraged by cyber actors to anonymize and amplify DDoS traffic.

Enterprises typically host various kinds of DNS assets. They will typically host a small number of recursive resolvers that proxy DNS requests from internal hosts to external DNS servers, and also cache results to reduce the number of external queries. Individual hosts may choose to over-ride the enterprise recursive resolvers, such as by manually changing their preferred resolver to a public one (such as Googles 8.8.8.8 and CloudFlares 1.1.1.1), but in general, a majority of hosts will use the default recursive resolver provided by their organization. In addition, enterprises typically host a number of authoritative name servers to serve the various domains belonging to the organization. For example, organization-wide services (like email, VPN, etc.) may be managed by central IT. At the same time, each department may operate its own authoritative name server to resolve department-specific web pages. It is not uncommon for the various IT entities to operate in silos, often unaware of the assets being managed by the other. To make matters worse, on-campus retail stores (bookshops, food outlets, etc.) that lease connectivity from the campus may also be housing their own DNS assets, which are often poorly secured as they lack the skills.

Distributed-Denial-of-Service (DDoS) attacks remain a serious problem, in spite of decades of research and commercial efforts to curb them. Ongoing Covid19 pandemic and increased reliance of our society on network services, have further increased opportunities for DDoS attacks. According to the security company F5 Labs, between January 2020 and March 2021, DDoS attacks have increased by 55% . While some large-volume DDoS attacks make front page news (for example, the 1.35 Tb/s attack on Github in Feb. 2018, or 2021 17.2 M requests per second attack, detected by CloudFlare), many more attacks occur daily and disrupt operations of thousands of targets.

Denial of Service (DoS) attacks pose a major, omnipresent threat to the stability of the Internet. About one-third of the active /24 networks on the Internet received DoS attacks over a two-year period , and 90% of attacks mitigated at a large IXP involved reflection attacks. To bring about reflection, attackers spoof source IP addresses to send request packets that supposedly originate from an intended victim, and abuse the infrastructure that replies to these requests (e.g., open DNS resolvers). Amplification is successful if the responses are larger than the requests. The DNS is a core Internet component. It primarily operates over the transport-layer protocol UDP. Due to its stateless nature, UDP is particularly susceptible to spoofing, and at least 14 protocols that work on top of UDP allow for reflection attacks. The Network Time Protocol (NTP) and DNS are (currently) the most-abused protocols.

Notably, amplification attacks are not limited to UDP. Poor implementations of network stacks allow attackers to use TCP as well. A recent DNS amplification attack exploits inefficient resolver implementations and works regardless of the underlying transport-layer protocol DNS amplification remains one of the most popular attack vectors, despite recent changes such as DNS-over-TLS and DNS-over-HTTPS.

A classic form of DDoS attacks still common in todays networks is the TCP SYN Attack. In this form of attack, the attacker initiates many TCP connections, while never completing the TCP handshake. The connection queue of the target is therefore filled up with incomplete connections, preventing it from addressing new connection requests from legitimate parties. The attacker may make an attack more difficult to detect by utilizing a botnet or a large army of sources for carrying out the attack or even by simply using spoofed sources. In this case, the attacked destination receives connection requests from many different sources.

Expert measurement methods are essential to observe global attack activities. Having a thorough understanding of attack dynamics and the abused infrastructure is crucial to effectively mitigate DNS-based attacks and to reduce the opportunity for infrastructure abuse. Several efforts exist to monitor amplification attacks on a global scale. Primarily, the monitoring infrastructures are implemented with the help of honeypots. In such works, careful assumptions are made about the share of global attacks that honeypots account for because the amplification ecosystem consists of a large number of amplifiers with high churn rates. Moreover, sophisticated attackers learn about the location of honeypots and exclude them.

There are multiple ways of describing the taxonomy and types of DDoS attacks, having different sources both from academia and business approaching the problem of taxonomy differently. Mirkovic and Reiher divide the different types in classes based on specific criteria that are both technical and social, e.g. based on the impact of the attack. Some of the criteria are degree of automation (DA), exploited weakness to deny service (EW), source address validity (SAV), attack rate dynamics (ARD), possibility of characterization (PC), persistence of agent set (PAS), victim type (VT) and impact on the victim (IV). On the other hand, the NIST provides only two types of DDoS attacks, flaw exploitation and flooding. The main distinction between the two types lies in the medium used to perform the attack and the end-system being attacked. In the flaw exploitation, the target is the software of the system, attempting to deplete its resources like memory, CPU, disk space or memory buffers. In flooding attacks, the attack is on the networking capabilities of the target, depleting the network capacity by accessing the resource with the means of attack, thus making it inaccessible for legitimate users.

Threats of malware, attacks and intrusion have been around since the very conception of computing. Yet, it was not until the sudden growth of the internet that awareness of security and digital assets really started to pick up steam. The internet presents a new liability, as the everincreasing number of machines on the web provides a new goldmine for those seeking to exploit vulnerabilities. As access increases, new ways are created for attackers to exploit network systems and their users. Among various types of attack, DDoS remains the most devastating and severe due to its potential impact, and the potentiality keeps on growing, making intrusion detection a must for network security and defense. As a result, machine learning and artificial intelligence research has flourished over the last few years, opening new doors for intrusion detection technologies. However, data availability still limits greatly the success of such technologies, as research faces a shortage of good quality IDS datasets.

During the last decades, our day-by-day life has been strictly connected to the usage of devices and online services, therefore making their efficiency and continuity play a crucial role in the technological transformation we witness. Likewise, the economic loss derived from cyberthreats has increased exponentially in recent years as the technologies continually evolve and attackers develop their skills. One of the most common ways cybercriminals try to jeopardize the continuity of systems and thus cause economic damage is Denial of Service (DoS), which aims to drain the computing capabilities of the target system in both fancy and basic ways. A case of this attack is the Distributed Denial of Service DDoS, where a network of infected devices (bots) are commanded by an attacker (botmaster) through a Command&Control Server (C&C). What happens to be erratic and thus detectable by a Machine Learning (ML) model in this kind of attack is the DNS traffic, carrying Domain Names through which bots are connected to the C&C server. This stealthy connection strategy is commonly known as Domain Fluxing, where the algorithms used by the infected bots to generate the domain are known as Domain Generation Algorithms (DGAs).

Although employing ML models to detect the presence of botnets within network traffic has been demonstrated to be successful, almost the entirety of the relevant works have followed a common baseline and workflow, presenting a partially novel feature set on which to train a classifier to obtain relevant results. The proposed approaches lack interpretability and contextualization. First, depending on the context from which DNS traffic data is extracted and the model is deployed, potential attackers might have control over some features. Second, the model prioritization and general usage of the features in the decision process are not known beforehand, making the process challenging to debug and protect.

Motivation

Motivated by a recent new type of randomized Distributed Denial of Service (DDoS) attacks on the Domain Name Service (DNS), we develop novel and efficient distinct deep algorithms and build an attack identification system that uses our algorithms.

Objectives

- How to infer large-scale DNS amplification DDoS activities?

- What are the characteristics of DNS amplification DDoS attacks?
- What inferences can we extract from analyzing DNS amplification DDoS traces?
- Which explain ability techniques can provide insight into how the model takes its decisions?
- How to train different classifiers and compare their performances?

Problem Statement

The relevance of the results achieved from such techniques rely on the quality of the datasets employed, as these are vital to have a realistic evaluation. The validity of current datasets has been thoroughly questioned in the cybersecurity space. It is a challenge for many researchers to find appropriate datasets to validate and test their methods and having a suitable dataset is a significant challenge itself. Privacy is a huge setback for availability of these datasets as they contain sensitive information. In the off chance that these are made available, they are heavily anonymized or obsolete. The unavailability of such datasets and the absence of certain statistical characteristics remains one of the major challenges for anomaly-based intrusion detection.

The nature of the datasets brings about data privacy issues due to certain security policies, the sensitivity of the data and the potential risk from disclosing such information. Moreover, there are other trust factors that inhibit realistic data from being shared among industry stakeholders and researchers. As a result, organisations often choose to not disclose the outcomes of computer attacks. Therefore, most researchers do not use realistic data when conducting their own studies

A thorough review of literature is carried out to analyze three key areas: the use of Deep learning in the context of intrusion detection; the state-of-the art of datasets, their characteristics and shortcomings; and, a review of recent works on the validity of existing datasets. This lays the groundwork for the problem definition and objective of this study, that is, to analyze the intrusion detection performance of DDoS datasets.

Domain Overview

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future.

Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative Adversarial Networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications.

Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data. Mathematics is useful for developing machine learning models and finally, computer science is used for implementing algorithms.

However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate results. Optimization techniques involve tuning the hyperparameters to reach an optimum result.

The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry.

Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms.

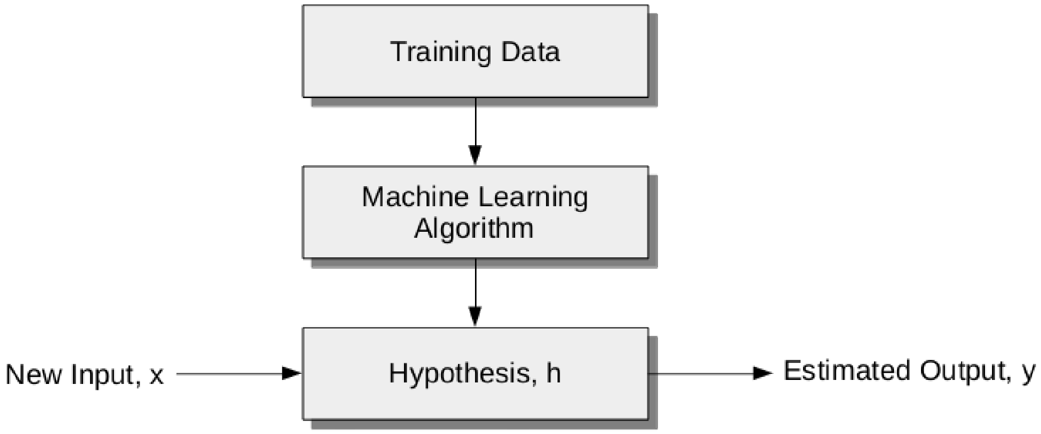
With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it.

These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions. Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

Types of Machine Learning

Machine Learning Algorithms can be classified into 3 types as follows:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning



Supervised Learning

In the majority of supervised learning applications, the ultimate goal is to develop a finely tuned predictor function h(x) (sometimes called the â€œhypothesisâ€). â€œLearningâ€ consists of using sophisticated mathematical algorithms to optimize this function so that, given input data x about a certain domain (say, square footage of a house), it will accurately predict some interesting value h(x) (say, market price for said house).

- **First Determine the type of training dataset**
- **Collect/Gather the labelled training data.**
- **Split the training dataset into training dataset, test dataset, and validation dataset.**
- **Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.**

- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

This form of analysis estimates the coefficients of the linear equation, involving one or more independent variables that best predict the value of the dependent variable. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. There are simple linear regression calculators that use a â€œleast squaresâ€ method to discover the best-fit line for a set of paired data. You then estimate the value of X (dependent variable) from Y (independent variable).

There are various selection methods for linear regression modeling in order to specify how independent variables are entered into the analysis. By using different methods, a variety of regression models from the same set of variables could be constructed. Forward variable selection enters the variables in the block one at a time based on entry criteria. Backward variable elimination enters all of the variables in the block in a single step and then removes them one at a time based on removal criteria. Stepwise variable entry and removal examines the variables in the block at each step for entry or removal. All variables must pass the tolerance criterion to be entered in the equation, regardless of the entry method specified. A variable is not entered if it would cause the tolerance of another variable already in the model to drop below the tolerance criterion6. In a model fitting the variables entered and removed from the model and various goodness-of-fit statistics are displayed such as R2, R squared change, standard error of the estimate, and an analysis-of-variance table.

1. Linear Regression
2. Regression Trees
3. Non-Linear Regression
4. Bayesian Linear Regression
5. Polynomial Regression
6.

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

1. Support Vector Classifier
2. Random Forest Classifier
3. Decision Tree Classifier
4. Ensemble Methods

About Pandas

Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The DataFrame is one of these structures.

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

Jupyter Notebooks offer a good environment for using pandas to do data exploration and modeling, but pandas can also be used in text editors just as easily.

Jupyter Notebooks give us the ability to execute code in a particular cell as opposed to running the entire file. This saves a lot of time when working with large datasets and complex transformations. Notebooks also provide an easy way to visualize pandasâ€™ DataFrames and plots. As a matter of fact, this article was created entirely in a Jupyter Notebook.

There are two types of data structures in pandas: Series and DataFrames.

1. Series: a pandas Series is a one dimensional data structure (â€œa one dimensional ndarrayâ€) that can store values â€” and for every value it holds a unique index, too.
2. DataFrame: a pandas DataFrame is a two (or more) dimensional data structure â€“ basically a table with rows and columns. The columns have names and the rows have indexes.

Numpy

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB, you might find this tutorial useful to get started with Numpy.

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

NumPy is, just like SciPy, Scikit-Learn, Pandas, etc. one of the packages that you just canâ€™t miss when youâ€™re learning data science, mainly because this library provides you with an array data structure that holds some benefits over Python lists, such as: being more compact, faster access in reading and writing items, being more convenient and more efficient.

NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for efficient computation of arrays and matrices. To work with these arrays, thereâ€™s a vast amount of high-level mathematical functions operate on these matrices and arrays.

An array is basically nothing but pointers. Itâ€™s a combination of a memory address, a data type, a shape, and strides:

- The data pointer indicates the memory address of the first byte in the array,
- The data type or dtype pointer describes the kind of elements that are contained within the array,
- The shape indicates the shape of the array, and
- The strides are the number of bytes that should be skipped in memory to go to the next element. If your strides are (10,1), you need to proceed one byte to get to the next column and 10 bytes to locate the next

row.

Mathplotlib

Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts. Plotting of graphs is a part of data vistualization, and this property can be achieved by making use of Matplotlib.

Matplotlib makes use of many general-purpose GUI toolkits, such as wxPython, Tkinter, QT, etc., in order to provide object-oriented APIs for embedding plots into applications. John D. Hunter was the person who originally wrote Matplotlib, and its lead developer was Michael Droettboom. One of the free and open-source Python library which is basically used for technical and scientific computing is Python SciPy. Matplotlib is widely used in SciPy as most scientific calculations require plotting of graphs and diagrams.

Matplotlib is a plotting library like GNUplot. The main advantage towards GNUplot is the fact that Matplotlib is a Python module. Due to the growing interest in python the popularity of matplotlib is continually rising as well.

Another reason for the attractiveness of Matplotlib lies in the fact that it is widely considered to be a perfect alternative to MATLAB, if it is used in combination with Numpy and Scipy. Whereas MATLAB is expensive and closed source, Matplotlib is free and open source code. It is also object-oriented and can be used in an object oriented way. Furthermore it can be used with general-purpose GUI toolkits like wxPython, Qt, and GTK+. There is also a procedural "pylab", which designed to closely resemble that of MATLAB. This can make it extremely easy for MATLAB users to migrate to matplotlib.

Matplotlib can be used to create publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Another characteristic of matplotlib is its steep learning curve, which means that users usually make rapid progress after having started. The officicial website has to say the following about this: "matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code."

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks.

Deep learning algorithms are constructed with connected layers.

- The first layer is called the Input Layer
- The last layer is called the Output Layer
- All layers in between are called Hidden Layers. The word deep means the network join neurons in more than two layers.

Scikit-learn

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

scikit-learn is an open source Machine Learning Python package that offers functionality supporting supervised and unsupervised learning. Additionally, it provides tools for model development, selection and evaluation as well as many other utilities including data pre-processing functionality.

Components of scikit-learn

1. Supervised learning algorithms
   Starting from Generalized linear models (e.g Linear Regression), Support Vector Machines (SVM), Decision Trees to Bayesian methods â€" all of them are part of scikit-learn toolbox. The spread of machine learning algorithms is one of the big reasons for the high usage of scikit-learn.

2. Cross-validation
   There are various methods to check the accuracy of supervised models on unseen data using sklearn.

3. Unsupervised learning algorithms
   There is a large spread of machine learning algorithms in the offering â€" starting from clustering, factor analysis, principal component analysis to unsupervised neural networks.

4. Various toy datasets
   This came in handy while learning scikit-learn.

5. Feature extraction
   Scikit-learn for extracting features from images and text (e.g. Bag of words)

Chapter 2: Literature Survey

Literature Survey 1	
Title	How Do Organizations Seek Cyber Assurance? Investigations on the Adoption of the Common Criteria and Beyond
Authors	Nan Sun , Chang-Tsun Li , Hin Chan , Md Zahidul Islam , Md Rafiqul Islam and Warren Armstrong
Published Year	2022
Efficiency	👍 Excellent empirical performance 👍 Improve the quality and consistency of data 👍 Minimizes the workload on infrastructures.
Drawbacks	👎 May take huge time and economic cost to construct. 👎 Maximizes the complexity of the problem 👎 Narrowly specialized knowledge
Description	Cyber assurance, which is the ability to operate under the onslaught of cyber attacks and other unexpected events, is essential for organizations facing inundating security threats on a daily basis. Organizations usually employ multiple strategies to conduct risk management to achieve cyber assurance. Utilizing cybersecurity standards and certifications can provide guidance for vendors to design and manufacture secure Information and Communication Technology (ICT) products as well as provide a level of assurance of the security functionality of the products for consumers. Hence, employing security standards and certifications is an effective strategy for risk management and cyber assurance. In this work, we begin with investigating the adoption of cybersecurity standards and certifications by surveying 258 participants from organizations across various countries and sectors. Specifically, we identify adoption barriers of the Common Criteria through the designed questionnaire. Taking into account the seven identified adoption barriers, we show the recommendations for promoting cybersecurity standards and certifications. Moreover, beyond cybersecurity standards and certifications, we shed light on other risk management strategies devised by our participants, which provides directions on cybersecurity approaches for enhancing cyber assurance in organizations. In this work, we presented the results of our survey on the Common Criteria adoption and approaches to ensuring cyber assurance for organizations. To determine if organizations have concerns related to cybersecurity regulatory issues as well as to determine organizations attitudes towards being measured against cybersecurity standards, seven adoption barriers of security standards and certifications are identified. The results of our study inform our recommendations for pro- moting Common Criteria adoption and broader cybersecurity standards and certifications. Aside from the use of cybersecu- rity standards and certifications to select secure ICT products, we investigate how organizations pursue cyber assurance and their adopted strategies. We hope the findings and recommen- dations we have made help researchers, organizations, and regulators raise concerns among academia and industry about the importance of cybersecurity standards and certifications. Beyond cybersecurity standards and certifications, the survey presents insights and directions on risk management, in the hope of inspiring organizations to achieve cyber assurance.

Literature Survey 2	
Title	A Survey on the Cyber Security of Small-to-Medium Businesses: Challenges, Research Focus and Recommendations
Authors	Alladean Chidukwani , Sebastian Zander and Polychronis Koutsakis
Published Year	2022
Efficiency	👍 Better operational efficiency 👍 Performs better on various circumstances and environment 👍 Improved reliability and resilience
Drawbacks	👎 Maximizes the complexity of the problem 👎 May take huge time and economic cost to construct. 👎 Resulting data errors.
Description	Small-to-medium sized businesses (SMBs) constitute a large fraction of many countries economies but according to the literature SMBs are not adequately implementing cyber security which leaves them susceptible to cyber-attacks. Furthermore, research in cyber security is rarely focused on SMBs, despite them representing a large proportion of businesses. In this paper we review recent research on the cyber security of SMBs, with a focus on the alignment of this research to the popular NIST Cyber Security Framework (CSF). From the literature we also summarise the key challenges SMBs face in implementing good cyber security and conclude with key recommendations on how to implement good cyber security. We find that research in SMB cyber security is mainly qualitative analysis and narrowly focused on the Identify and Protect functions of the NIST CSF with very little work on the other existing functions. SMBs should have the ability to detect, respond and recover from cyber-attacks, and if research lacks in those areas, then SMBs may have little guidance on how to act. Future research in SMB cyber security should be more balanced and researchers should adopt well-established powerful quantitative research approaches to refine and test research whilst governments and academia are urged to invest in incentivising researchers to expand their research focus. Continuous on-going research is required to support the development of cyber security solutions for SMBs [15], [102]. Research in cyber security is however rarely focussed on SMBs, despite them representing a large proportion of business. SMBs contribute immensely to the global economy, and in particular in Australia they make up 98% of all businesses contributing one third of the GDP. Despite their large number and importance, our study shows that research in SMB cyber security is rather limited and narrowly focussed. This is consistent with previous ndings of other researchers [15]. We also found SMB cyber security research to be con- centrated in the USA despite other nations having similar high proportions of SMBs and facing similar threats but in different environments.

Literature Survey 3	
Title	Analyzing and Evaluating Critical Cyber Security Challenges Faced by Vendor Organizations in Software Development: SLR Based Approach
Authors	Abdul Wahid Khan , Shah Zaib , Faheem Khan , Ilhan Tarimer , Jung Taek Seo and Jiho Shin
Published Year	2022
Efficiency	👍 Computational Complexity is significantly reduced 👍 Eliminating the huge workload of traditional methods 👍 May not meet the real-time requirement.
Drawbacks	👎 Big payloads 👎 Unsuitable for large scale scenarios. 👎 High complexity of installing and maintaining
Description	Security is the protection from various kinds of threats and most organizations engage in the challenge of security especially cyber-attacks. The attacks are increasing rapidly, due to which cyber security does not now change on supervised and pattern-based detection algorithms which assure continuous security observing. There are many kinds of problems in vendor organizations like cyber theft, which is the most common attack in cyberspace. This research study is developing a Cyber Security Challenges Model (CSCM) that will facilitate vendors organizations to identify challenges of cyber security during the development of software in a vendor organization. To find cyber security issues/challenges, a Systematic Literature Review (SLR) is conducted on 44 relevant research publications by developing a search string based on research questions. As the final selected research publications were less in number and did not complete our aim, therefore, snow bowling technique is applied to 67 relevant research publications. This relevant data was comprised of different databases/sources e.g., Google Scholar, IEEE Explore, SpringerLink, ACM Digital Library, anFffid ScienceDirect. Furthermore, for the distinctive literature review, weve carried out all of the steps in SLR, for example, improvement of SLR protocol, initials, and a very last collection of the applicable information, data extraction, data quality assessment, and data synthesis. Thirteen (13) critical cyber security challenges are identified which are; Security issues/Access of Cyberattacks, Lack of Right Knowledge, Framework, Lack of Technical Support, Disaster Issues, Cost Security issues, Lack of Confidentiality and Trust, Lack of Management, Unauthorized Access issues, Lack of Resources, Lack of Metrics, Administrative Mistakes during Development and Lack of Quality, Liability, and Reliability. The findings of our analysis study signify the similarities and dissimilarities in the recognized cybersecurity challenges in different decades, companies/firms, continents, databases, and methodologies. Through SLR, we have identied a list of 13 challenges which are all marked as critical challenges for vendor orga- nization during software development in CSCM as shown in (48%), D (43%), E (40%), F (39%), G (37%), H (33%), I (31%), J (28%), K (28%), L (27%) and M (25%). The vendor organizations need to give proper attention to these critical challenges to avoid any risk of failure by addressing these challenges.

Literature Survey 4	
Title	Last Line of Defense: Reliability Through Inducing Cyber Threat Hunting With Deception in SCADA Networks
Authors	Abdul Basit Ajmal , Masoom Alam , Awais Abdul Khaliq , Shawal Khan , Zakria Qadir and M. A. Parvez Mahmud
Published Year	2021
Efficiency	👍 Quick and Efficient to use 👍 Delivering information on the state of operation. 👍 Fast and efficient, but also as accurate as the state-of-the-art algorithms
Drawbacks	👎 High complexity of installing and maintaining 👎 They are hard to maintain 👎 Complexity of its Real Time Implementation
Description	There exists a gap between existing security mechanisms and their ability to detect advancing threats. Antivirus and EDR (End Point Detection and Response) aim to detect and prevent threats; such security mechanisms are reactive. This approach did not prove to be effective in protecting against stealthy attacks. SCADA (Supervisory Control and Data Acquisition) security is crucial for any country. However, SCADA is always an easy target for adversaries due to a lack of security for heterogeneous devices. An attack on SCADA is mainly considered a national-level threat. Recent research on SCADA security has not considered unknown threats, which has left a gap in security. The proactive approach, such as threat hunting, is the need of the hour. In this research, we investigated that threat hunting in conjunction with cyber deception and kill chain has countervailing effects on detecting SCADA threats and mitigating them. We have used the concept of decoy farm in the SCADA network, where all attacks are engaged. Moreover, we present a novel threat detection and prevention approach for SCADA, focusing on unknown threats. To test the effectiveness of approach, we emulated several SCADA, Linux and Windows based attacks on a simulated SCADA network. We have concluded that our approach detects and prevents the attacker before using the current reactive approach and security mechanism for SCADA with enhanced protection for heterogeneous devices. The results and experiments show that the proposed threat hunting approach has significantly improved the threat detection ability. Due to the change of threat landscape, reactive approaches are ineffective in detecting and reacting in time, resulting in no detection or increasing duel time between incident response and attack. Proactive approaches in conjunction with decep- tion and threat intelligence are an effective way of detect- ing and preventing threats quickly and using SCADA decoy farm to engage in attack and record its activity by providing IOCs to threat hunters. Hence we concluded that the threat detection ability of SCADA is increased using the threat hunting approach against real-world attacks as compared to traditional security mechanisms. For future directions, Our future work includes Adversary simulation on networks to mature our threat hunting teams with regular adversary exercises.

Literature Survey 5	
Title	A Cyber Security Evaluation Framework for In-Vehicle Electrical Control Units
Authors	Haichun Zhang , Yuqian Pan , Zhaojun Lu , Jie Wang and Zhenglin Liu
Published Year	2021
Efficiency	👍 Simplify the implementation process. 👍 Effectiveness for distributed optimization 👍 Achieve a well-balanced tradeoff among various parameters.
Drawbacks	👎 Generally have high polynomial running times. 👎 Narrowly specialized knowledge 👎 It is not an easy-to-use method
Description	Modern vehicles are equipped with more than 100 Electrical Control Units (ECUs) with over 2500 signals to transmit internally. The application of advanced electronics and communication techniques helps a vehicle transform from an information island into a powerful distribution center. However, a large number of ECUs have introduced a wider range of security threats for vehicles. The attackers can compromise a vehicle remotely through a vulnerable ECU. How to evaluate the cyber security of in-vehicle ECUs has become an important issue. Current Threat Analysis and Risk Assessment (TARA) only carries out theoretical analysis on the potential threats and risks faced by the vehicle in the conceptual design phase of the lifecycle, but lacks the details of actual security evaluation. In this paper, we proposed a Cyber Security Evaluation Framework (CSEF) to independently evaluate the security of the in-vehicle ECUs, which is composed of the asset identification, the threat analysis, the risk assessment, and the security test. The proposed CSEF is applied to a pre-installed On-Bord Unit (OBU) to provide a use case. The use case show that the proposed CSEF is able to figure out assets, threats, risks behind threats, and vulnerabilities of OBU, playing an important role in guiding others to conduct security evaluation. Moreover, CSEF can be extended to evaluate the cyber security of other critical ECUs, such as the Telematic Box, the infotainment units, and the gateway. In order to better apply the security assessment to the IoV, we analyzed the security architecture of the IoV in detail, and proposed a security framework for the IoV. The security framework focuses on ten security aspects of smart vehicles and in-vehicle ECUs at four levels, which regulates the scope of security evaluation. In addition, we proposed CSEF that can be applied to in-vehicle ECUs to evaluate the cyber security of in-vehicle ECUs.The CSEF is designed based on the ISO/SAE 21434 standard and is optimized to have richer security assessment details, which can be better applied to the field of automotive security. The framework aims to solve five main problems:(1) identifies the assets of the evaluated objectives from ten aspects.

Literature Survey 6	
Title	A Review of Security Standards and Frameworks for IoT-Based Smart Environments
Authors	Nickson M. Karie , Nor Masri Sahri , Wencheng Yang , Craig Valli and Victor R. Kebande
Published Year	2021
Efficiency	👍 Effectiveness for distributed optimization 👍 Ability To Deliver High Quality Results 👍 Capable of further reducing the required level of human effort
Drawbacks	👎 They are hard to maintain 👎 Prone to Errors 👎 Generally have high polynomial running times.
Description	Assessing the security of IoT-based smart environments such as smart homes and smart cities is becoming fundamentally essential to implementing the correct control measures and effectively reducing security threats and risks brought about by deploying IoT-based smart technologies. The problem, however, is in finding security standards and assessment frameworks that best meets the security requirements as well as comprehensively assesses and exposes the security posture of IoT-based smart environments. To explore this gap, this paper presents a review of existing security standards and assessment frameworks which also includes several NIST special publications on security techniques highlighting their primary areas of focus to uncover those that can potentially address some of the security needs of IoT-based smart environments. Cumulatively a total of 80 ISO/IEC security standards, 32 ETSI standards and 37 different conventional security assessment frameworks which included 7 NIST special publications on security techniques were reviewed. To present an all-inclusive and up-to-date state-of-the-art research, the review process considered both published security standards and assessment frameworks as well as those under development. The findings show that most of the conventional security standards and assessment frameworks do not directly address the security needs of IoT-based smart environments but have the potential to be adapted into IoT-based smart environments. With this insight into the state-of-the-art research on security standards and assessment frameworks, this study helps advance the IoT field by opening new research directions as well as opportunities for developing new security standards and assessment frameworks that will address future IoT-based smart environments security concerns. This paper also discusses open problems and challenges related to IoT-based smart environments security issues. As a new contribution, a taxonomy of challenges for IoT-based smart environment security concerns drawn from the extensive literature examined during this study is proposed in this paper which also maps the identified challenges to potential proposed solutions.

Literature Survey 7	
Title	Offensive Security: Towards Proactive Threat Hunting via Adversary Emulation
Authors	Abdul Basit Ajmal , Munam Ali Shah , Carsten Maple , Muhammad Nabeel Asghar and Saif Ul Islam
Published Year	2021
Efficiency	👍 Simplify the implementation process. 👍 Better operational efficiency 👍 Keeping the control overhead at regular levels
Drawbacks	👎 Faces critical design challenges. 👎 Heavyweight 👎 Approach is a bit time-consuming
Description	Attackers increasingly seek to compromise organizations and their critical data with advanced stealthy methods, often utilising legitimate tools. In the main, organisations employ reactive approaches for cyber security, focused on rectifying immediate incidents and preventing repeat attacks, through protections such as vulnerability assessment and penetration testing (VAPT) security information and event management (SIEM), firewalls, anti-spam/anti-malware solutions and system patches. Such system have weaknesses in addressing modern modern stealthy attacks. Proactive approaches, have been seen as part of the solution to this problem. However, approaches such as VAPT have limited scope and only works with threats that have already been discovered. Promising methods such as threat hunting are gaining momentum, enabling organisations to identify and rapidly respond to any potential attacks, though they have been criticised for their significant cost. In this paper, we present a novel hybrid model for uncovering tactics, techniques, and procedures (TTPs) through offensive security, specifically threat hunting via adversary emulation. The proposed technique is based on a novel approach of inducing adversary emulation (mapping each respective phase) model inside the threat hunting approach. The experimental results show that the proposed approach uses threat hunting via adversary emulation and has countervailing effects on hunting advance level threats. Moreover, the threat detection ability of the proposed approach utilizes minimum resources. The proposed approach can be used to develop the offensive security-aware environment for organizations to uncover advanced attack mechanisms and test their ability for attack detection. This paper has presented a novel hybrid model for launch- ing offensive security exercises to capture, determine and understand attack patterns by foreseen threats using threat hunting. The proposed approach has increased the efciency of identifying and countering threats using real world attack scenarios and presents an algorithm to generate attack vectors for phishing. In contrast to traditional methods that focus on known threats, such as VAPT. The proposed scheme is designed to identify and address emerging unknown threats. In the future, we plan to focus on increasing the realism of the emulation of adversaries with advanced stealthy attacks.

Literature Survey 8	
Title	Cyber Threat Predictive Analytics for Improving Cyber Supply Chain Security
Authors	Abel Yeboah-Ofori , Shareeful Islam , Sin Wee Lee , Zia Ush Shamszaman , Khan Muhammad , Meteb Altaf and Mabrook S. Al-Rakhami
Published Year	2021
Efficiency	👍 Quick and Efficient to use 👍 Increased efficiency and speed 👍 Low Deployment Cost
Drawbacks	👎 Difficulties to obtain better performance 👎 High complexity of installing and maintaining 👎 Cannot be implemented real time
Description	Cyber Supply Chain (CSC) system is complex which involves different sub-systems performing various tasks. Security in supply chain is challenging due to the inherent vulnerabilities and threats from any part of the system which can be exploited at any point within the supply chain. This can cause a severe disruption on the overall business continuity. Therefore, it is paramount important to understand and predicate the threats so that organization can undertake necessary control measures for the supply chain security. Cyber Threat Intelligence (CTI) provides an intelligence analysis to discover unknown to known threats using various properties including threat actor skill and motivation, Tactics, Techniques, and Procedure (TT and P), and Indicator of Compromise (IoC). This paper aims to analyse and predicate threats to improve cyber supply chain security. We have applied Cyber Threat Intelligence (CTI) with Machine Learning (ML) techniques to analyse and predict the threats based on the CTI properties. That allows to identify the inherent CSC vulnerabilities so that appropriate control actions can be undertaken for the overall cybersecurity improvement. To demonstrate the applicability of our approach, CTI data is gathered and a number of ML algorithms, i.e., Logistic Regression (LG), Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT), are used to develop predictive analytics using the Microsoft Malware Prediction dataset. The experiment considers attack and TTP as input parameters and vulnerabilities and Indicators of compromise (IoC) as output parameters. The results relating to the prediction reveal that Spyware/Ransomware and spear phishing are the most predictable threats in CSC. We have also recommended relevant controls to tackle these threats. We advocate using CTI data for the ML predicate model for the overall CSC cyber security improvement. The integration of complex cyber physical infrastructures and applications in a CSC environment have brought eco- nomic, business, and societal impact for both national and global context in the areas of Transport, Energy, Healthcare, Manufacturing, and Communication. However, CPS security remains a challenge as vulnerability from any part of the system can pose risk within the overall supply chain context. This paper aims to improve CSC security by integrating CTI and ML for the threat analysis and predication.

Literature Survey 9	
Title	Factors Related to Cyber Security Behavior
Authors	Ana Kovaevi , Nenad Putnik and Oliver Tokovi
Published Year	2020
Efficiency	👍 Found attractive outcomes 👍 Provides the integrity and nontransferablity. 👍 Relatively simple and computationally inexpensive method
Drawbacks	👎 High complexity of installing and maintaining 👎 Heavyweight 👎 Solutions have been proved ineffective
Description	Theoretical and empirical insight notes that cyber security awareness is a topic of particular interest in cyber security. Humans are the central figures in cyber security and the way to reduce risk in cyberspace is to make people more security aware. While there have been numerous studies about various aspects of cyber security awareness, they are both inconsistent and environment-dependent. The main aim of our research is to analyze cyber security awareness in depth, and to try to discover how various factors such as socio-demographics, cyber security perceptions, previous cyber security breaches, IT usage, and knowledge may individually or together impact on cyber security behavior. To prove that we conducted our research on students, as they are the most technologically active part of the society. We discovered that knowledge proved to be the dominant factor for cyber security awareness, and although students are digital natives, they do not feel safe in the cyber environment; they do not behave securely and do not have adequate knowledge to protect themselves in cyberspace. The environment is a very important factor when analyzing cyber security, as stated in [15], and this is the first sur- vey conducted among university students in Serbia (in par- ticular freshmen), which analyzes the factors relevant for cyber security awareness in depth. In addition, our survey also analyzed unreported correlations; how various factors in A. Kovaevi et al.: Factors Related to Cyber Security Behavior particular and together, such as socio-demographic character- istics, cyber security perceptions, cyber security breach expe- riences, IT usage, and knowledge influence security behavior. It was shown that the effects of cyber security perceptions, knowledge, and experiences are stronger than the effects of socio-demographics for cell phone related behavior, or in particular, IT usage and knowledge appeared as significant predictors of cell phone related behavior. However, any sig- nificant predictors have not been discovered for password related behavior, which will be the focus of our future anal- ysis. Even though our participants perceived that their data were not safe, this did not serve as a trigger for them to learn more about cyber security so as to find out how to behave more securely in cyberspace.

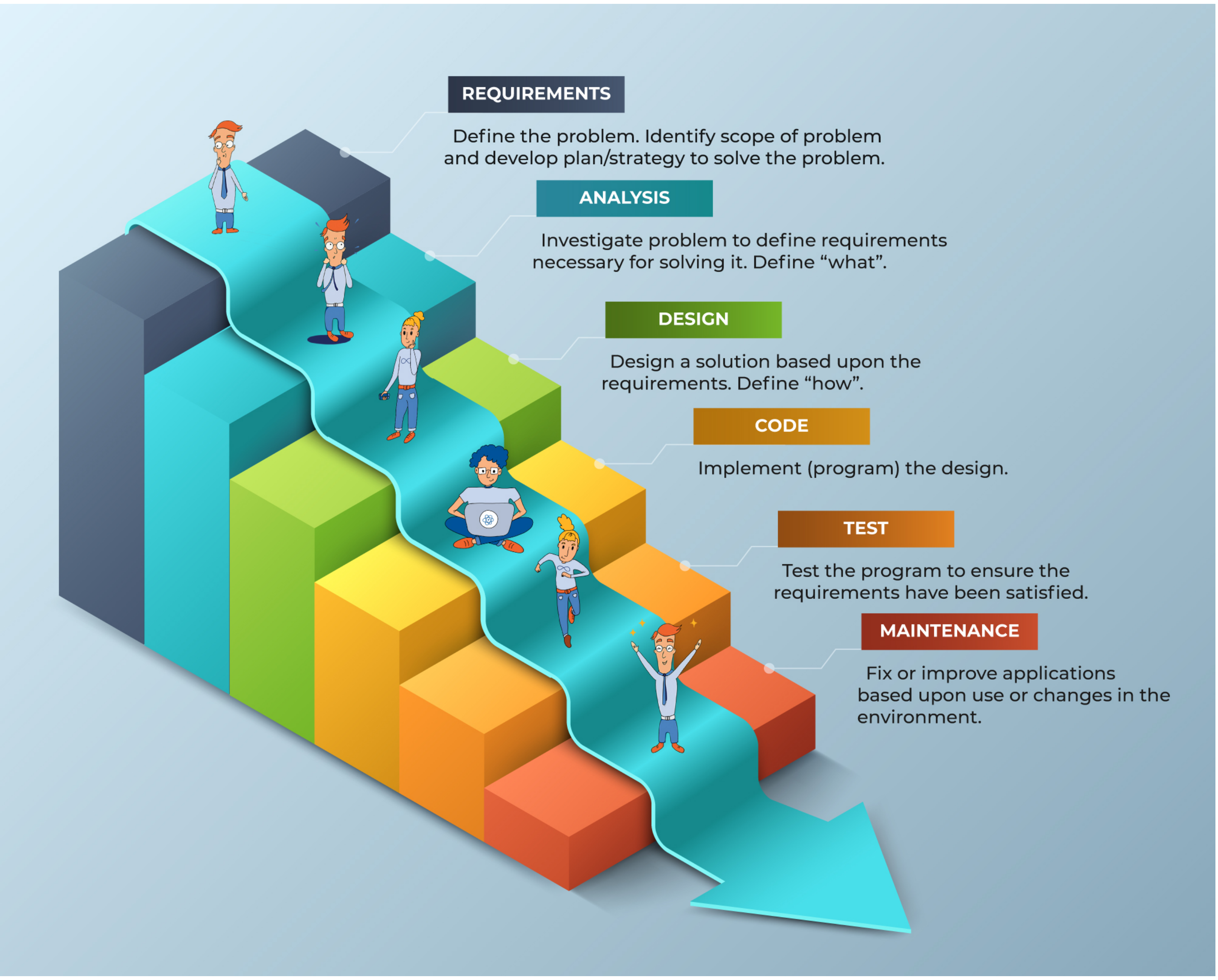
Literature Survey 10	
Title	A Risk Management Approach to Double-Virus Tradeoff Problem
Authors	Jichao Bi , Fangfei Zhang , Ali Dorri , Chunming Zhang and Chen Zhang
Published Year	2019
Efficiency	👍 Simplify the implementation process. 👍 Fast and efficient, but also as accurate as the state-of-the-art algorithms 👍 Lowering the Complexity Threshold
Drawbacks	👎 Approach is a bit time-consuming 👎 Resulting data errors. 👎 Generally have high polynomial running times.
Description	One of the major threats to cybersecurity is the emergence of new computer viruses. By emergence of new viruses, the cybersecurity companies assign a team of security experts and programmers to study the behavior of the virus and develop the corresponding antivirus program to secure networks. Sometimes, more than one new virus is identified that requires the cybersecurity team to make a tradeoff on the allocation of programmers, and thus leads to two-antivirus-program development problem under double-virus attack. In this paper, we propose DOWNHILL algorithm to address the outlined challenge. We model the time evolution of the expected state of the network as a differential dynamical system to measure the total loss caused by the viruses. Then, we propose a DOWNHILL algorithm, three heuristic algorithms and a random algorithm to solve the problem, respectively. We study the computational complexities of the proposed algorithms as well. Through numerous comparative experiments, we confirm the DOWNHILL algorithm is the most effective method to this problem. Finally, the influence of different factors on the DOWNHILL strategy and its potential total loss are also researched. In the framework of risk management, this paper studied the computer virus response problem. On this basis, we modeled the problem of tackling with new viruses as a two-antivirus- program development problem. The latter is defined as how to allocate the limited programmer resources to develop dif- ferent antivirus programs to achieve the minimum potential total loss (caused by viruses). Through modeling and anal- ysis of the time evolution of the networks expected state, we quantified the networks potential loss caused by Virus I and II, respectively. Then, we obtained the potential total loss by adding them together. We presented DOWNHILL algorithm to solve this problem. The simulation results prove that the proposed DOWNHILL algorithm achieves better performance as compared to three heuristic algorithms and the random algorithm. We also studied the impact of multiple factors on DOWNHILL strategy. To the best of our knowl- edge, this work is the first work that studies the double-virus tradeoff problem.

Chapter 3: System Analysis

Waterfall Model

It is called as traditional approach. Waterfall is a linear method (sequential model) for any software application development. Here application development is segregated into a sequence of pre -defined phases.

Waterfall Model (Taken from Google.com)


REQUIREMENTS
Define the problem. Identify scope of problem and develop plan/strategy to solve the problem.
ANALYSIS
Investigate problem to define requirements necessary for solving it. Define "what".
DESIGN
Design a solution based upon the requirements. Define "how".
CODE
Implement (program) the design.
TEST
Test the program to ensure the requirements have been satisfied.
MAINTENANCE
Fix or improve applications based upon use or changes in the environment.

1. **Requirement gathering and documentation**
   In this stage, you should gather comprehensive data approximately what this challenge requires. You may gather this data in a diffusion of ways, from interviews to questionnaires to interactive brainstorming. By means of the stop of this section, the task requirements have to be clean, and also you have to have a necessities file that has been allotted on your team.

2. **System design**
   Using the well-known requirements, your team designs the solutions. During this phase, no development will be happening. But the project team starts specification such as programming language or hardware

requirements.

3. Implementation

   During this phase software development coding will be happening. Web application programmers take data from the previous stage and create a functional product. Web application programmers write source code in small pieces, which are integrated at the end of this phase or the beginning of the next.

4. Testing

   Once all coding is done, testing of the product can begin. Testers methodically find and report any problems. If serious issues arise, your project may need to return to phase one for revaluation.

5. Delivery/deployment

   In this phase, the solution is complete, and your project team submits the deliverables to be deployed or released.

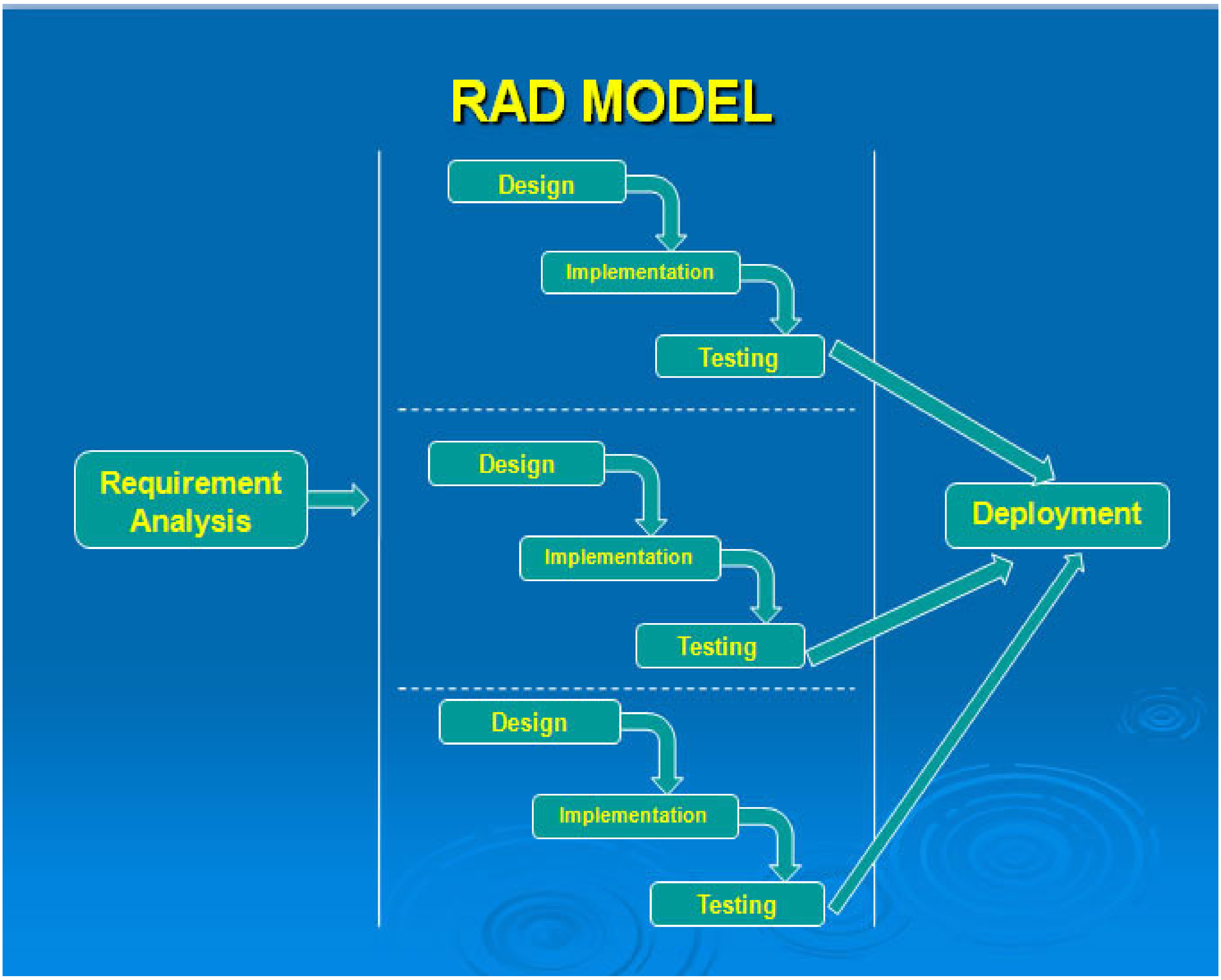
6. Maintenance

   The final solution has been implemented to the client and is being used. As troubles arises, the project team might also want to create patches and updates might also to deal with them. Again, huge troubles may also necessitate a return to segment one.

RAD Model

Rapid application development is an agile software development approach that focuses more on ongoing software projects and user feedback and less on following a strict plan. As such, it emphasizes rapid prototyping over costly planning.

RAD Model (Taken from Google.com)



1. Define Requirements

   Rather than making you spend months developing specifications with users, RAD begins by defining a loose set of requirements. â€œLooseâ€ because among the key principles of rapid application development is the permission to change requirements at any point in the cycle.

   Basically, developers gather the products gist. The client provides their vision for the product and comes to an agreement with developers on the requirements that satisfy that vision.

   This phase is equivalent to a project scoping meeting. Although the planning phase is condensed compared to other project management methodologies, this is a critical step for the ultimate success of the project.

   During this stage, web application programmers, clients (software users), and team members communicate to determine the goals and expectations for the project as well as current and potential issues that would need to be addressed during the build.

   1. Researching the current problem
   2. Defining the requirements for the project
   3. Finalizing the requirements with each stakeholders approval

   A basic breakdown of this stage involves:

1. It is important that everyone has the opportunity to evaluate the goals and expectations for the project and weigh in. By getting approval from each key stakeholder and web application programmer, teams can avoid miscommunications and costly change orders down the road.

2. Prototype
   In this rapid application development phase, the developerâ€™s goal is to build something that they can demonstrate to the client. This can be a prototype that satisfies all or only a portion of requirements (as in early-stage prototyping).

   This prototype may cut corners to reach a working state, and thatâ€™s acceptable. Most RAD programming approaches have a finalization stage where developers pay down technical debt accrued by early prototypes.

   All the bugs and kinks are worked out in an iterative process. The web application programmer designs a prototype, the client (user) tests it, and then they come together to communicate on what worked and what did not.

   This method gives web application programmers the opportunity to tweak the model as they go until they reach a satisfactory design. Both the software web application programmers and the clients learn from the experience to make sure there is no potential for something to slip through the cracks.

3. Absorb Feedback
   With a recent prototype prepared, RAD developers present their work to the client or end-users. They collect feedback on everything from interface to functionalityâ€"it is here where product requirements might come under scrutiny.

   Clients may change their minds or discover that something that seemed right on paper makes no sense in practice. Clients are only human, after all. With feedback in hand, developers return to some form of step 2: they continue to prototype. If feedback is strictly positive, and the client is satisfied with the prototype, developers can move to step 4.

   Because the majority of the problems and changes were addressed during the thorough iterative design phase, web application programmers can construct the final working model more quickly than they could by following a traditional project management approach.

   The phase breaks down into several smaller steps:

   1. Preparation for rapid construction
   2. Program and application development
   3. Coding
   4. Unit, integration, and system testing

   The software development team of programmers, coders, testers, and web application programmers work together during this stage to make sure everything is working smoothly and that the end result satisfies the clients expectations and objectives. This third phase is important because the client still gets to give input throughout the process. They can suggest alterations, changes, or even new ideas that can solve problems as they arise.

4. Finalize Product
   During this stage, developers may optimize or even re-engineer their implementation to improve stability and maintainability. They may also spend this phase connecting the back-end to production data, writing thorough documentation, and doing any other maintenance tasks required before handing the product over with confidence.

Existing System

This paper investigates the mathematical modelling of cybercrime attacks on multiple devices connected to the server. This model is a very successful way for cybercrime, bio-mathematics, and artificial intelligence to investigate and comprehend the behaviour of mannerisms with harmful intentions in a computer system. In this computational model, the existing system authors are studying the factors (i.e., computer viruses, disease infections, and cyberattacks) that affect connected devices. This compartmental model, SEIAR, represents the various hardware utilised during the cyberattack. The letters S, E, I, A, and R are used to represent different stages or groups of individuals in epidemiological models, helping to understand the spread and control of infectious diseases. The dynamics of the previous model are determined by a series of differential equations. The dynamics of the preceding model are determined by a system of differential equations. Numerical solutions of the model are calculated using backpropagated Levenberg-Marquardt algorithm (BLMA) and a specific optimization algorithm known as the Levenberg-Marquardt algorithm (LMA). Reference solutions were obtained by using the Runge-Kutta algorithm of order 4 (RK-4). The backpropagated Levenberg-Marquardt algorithm (BLMA), commonly known as the damped least-squares (DLS) method. Subsequently, the existing system authors endeavor to analyze the surrogate solutions obtained for the system and determine the stability of our approach. Moreover, the existing system authors aim to ascertain fitting curves to the target solutions with minimum errors and achieve a regression value of 1 for all the predicted solutions. The outcome of our simulations ensures that our approach is capable of making precise predictions concerning the behavior of real-world phenomena under varying circumstances. The testing, validation, and training of our technique concerning the reference solutions are then used to determine the accuracy of the surrogate solutions obtained by BLMA. Convergence analysis, error histograms, regression analysis, and curve fitting were used for each differential equation to examine the robustness and accuracy of the design strategy.

In this work, the existing system authors use one of the intelligent techniques based on an artificial neural network to investigate the mathematical model that simulates Pony Stealer (malware attack) in the connection that has been developed. The mathematical model is compartmental since asymptomatic devices, as well as Exposed Susceptible, Susceptible, Infectious, and Recovered, have all been regarded as separate systems linked by a single server. Some infections can propagate through asymptomatic devices without causing symptoms. These viruses are identified through infectious devices. This extra type of device is crucial to include in cyber security models since many cyberattacks are intended to control the device system in an anonymous manner in order to collect personal data [68]. Such real-world processes are regulated by a set of ordinary differential equations. Deep neural learning-based machine learning techniques [69], have been applied to solve the system of ordinary differential equations underlying the epidemic model. In the ANN approach, the existing system authors use one hidden layer for sample points of each equation in Matlab, and using the RK-4 approach, a reference solution is generated, which is later analysed using the Levenberg-Marquardt algorithms training, testing, and validation procedures. Since the approximate solutions and analytical answers correspond with the lowest absolute errors when compared to state-of- the-art techniques, the detailed graphical analysis shows that the suggested method is accurate and effective. Additionally, performance indicator values are getting closer to zero, demonstrating flawless outcome modelling. VII. CONFLICTS OF INTEREST The author declare no conflicts of interest.

Drawbacks of Existing System

- Narrowly specialized knowledge
- Cannot meet current network business demands
- High complexity, inaccuracy, and inadequacy

- High complexity of installing and maintaining
- Makes fine-grained source-IP filtering much harder.

Proposed System

Our goal is to design an AI brain, which continuously evaluates suitability of multiple filters to handle an ongoing DDoS attack on a DNS root server. Our system needs to quickly select the best filter or the combination of filters, reasoning about the projected impact on the attack, the collateral damage from the filter on legitimate recursive traffic and the operational cost. The system should also be able to adjust its selection as attack changes. Finally, individual filters need to be configured to achieve optimal performance high effectiveness against attacks they are designed to detect and low collateral damage.

Feature Statistical Analysis. These plots show the marginal distributions of every pair of features as density plots, describing how the distributions for the classes behave. Through the scattered plots instead, we can assess where both benign and malicious samples lie in their adhoc feature space, thus making us capable of understanding to which extent pairs of features separate the data. Analyzing the scattered plots allows observing the distribution of the features to get a rough idea of how they will behave/discriminate and to which extent.

Features considered to incorporate time series elements of the attacks included averaging descriptive statistics such as minimum or maximum of different components, e.g., labels or prefix lengths, per minute over an attack, as well as variance and co-variance measures. While promising, these features proved more difficult to assess and scale given the volume of data and were not used in the final clustering. Given the strong time elements in DDoS attacks, improved approaches to leveraging time series features at scale would likely prove valuable to understanding the malware and actors.

We propose a method of classifying a DNS server, according to whether or not it is used as a reflector, by monitoring the incoming DNS messages. We collect a series of DNS packets sent from a DNS server and build a feature matrix of the server, assuming that a reflector may have a different packet sequence pattern than that found with a normal DNS server. The preliminary result shows that our method can classify reflectors with an F1 score greater than 0.9 when the test and training data are generated within the same day. The trained model can also classify the data not used for the training and testing phase of the same day with more than 0.7 F1 score.

Advantages of Proposed System

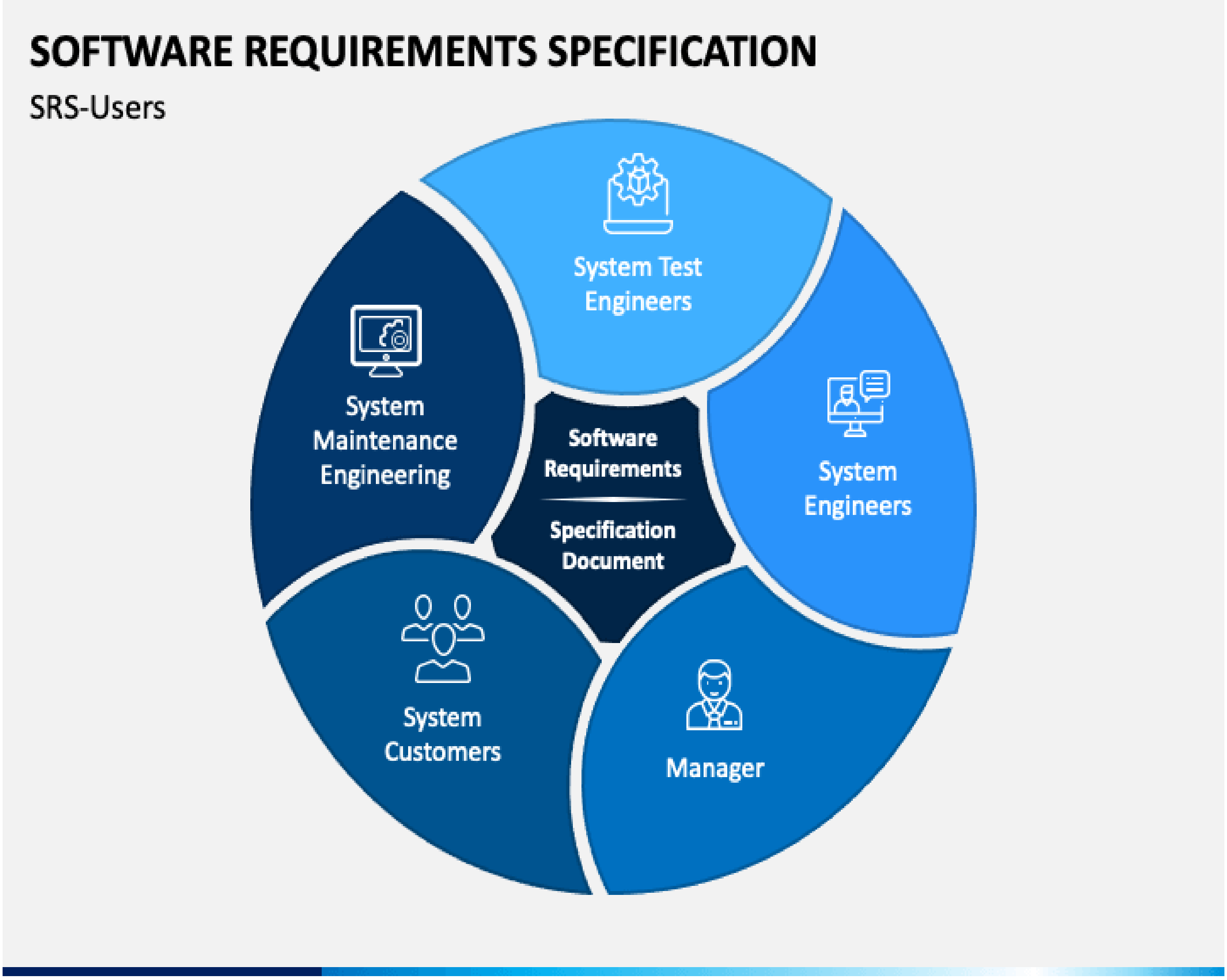
- Its not difficult to see what is Impacted
- Simple, fast and less complex.
- Excellent empirical performance
- Streamlined and decoupled services
- Simple to use and interpret

Chapter 4: Requirement Specification

Introduction

Clearly defined requirements are essential signs on the road that leads to a successful project. They establish a formal agreement between a client and a provider that they are both working to reach the same goal. High-quality, detailed requirements also help mitigate financial risks and keep the project on a schedule. According to the Business Analysis Body of Knowledge definition, requirements are a usable representation of a need.

Creating requirements is a complex task as it includes a set of processes such as elicitation, analysis, specification, validation, and management.



A System Requirements Specification (SRS) (also known as a Software Requirements Specification) is a document or set of documentation that describes the features and behavior of a system or software application.

Depending on the methodology employed (agile vs waterfall) the level of formality and detail in the SRS will vary, but in general an SRS should include a description of the functional requirements, system requirements, technical requirements, constraints, assumptions and acceptance criteria. Each of these is described in more detail below:

- Business Drivers
- Business Model
- Functional and System Requirements
- Business and System Use Cases
- Technical Requirements
- System Qualities
- Constraints and Assumptions
- Acceptance Criteria

Business Drivers

This section describes the reasons why the customer is looking to build the system. The rationale for the new system is important as it will guide the decisions made by the business analysts, system architects and developers. Another compelling reason for documenting the business rationale behind the system is that the customer may change personnel during the project. Documentation which clearly identifies the business reasons for the system will help sustain support for a project if the original sponsor moves on.

The drivers may include both problems (reasons why the current systems/processes are not sufficient) and opportunities (new business models that the system will make available). Usually a combination of problems and opportunities are needed to provide motivation for a new system.

Business Model

This section describes the underlying business model of the customer that the system will need to support. This includes such items as the organizational context, current-state and future-state diagrams, business context, key business functions and process flow diagrams. This section is usually created during the functional analysis phase.

Functional and System Requirements

This section usually consists of a hierarchical organization of requirements, with the business/functional requirements at the highest-level and the detailed system requirements listed as their child items.

Business and System Use Cases

This section usually consists of a UML use case diagram that illustrates the main external entities that will be interacting with the system together with the different use cases (objectives) that they will need to carry out. For each use-case there will be formal definition of the steps that need to be carried out to perform the business objective, together with any necessary pre-conditions and post-conditions.

The business use cases are usually derived from the functional requirements and the system use cases are usually derived from the system requirements.

Technical Requirements

This section is used to list any of the "non-functional" requirements that essentially embody the technical environment that the product needs to operate in, and include the technical constraints that it needs to operate under. These technical requirements are critical in determining how the higher-level functional requirements will get decomposed into the more specific system requirements.

System Qualities

This section is used to describe the "non-functional" requirements that define the "quality" of the system. These items are often known as the "-ilities" because most of them end in "ility". They included such items as: reliability,

availability, serviceability, security, scalability, maintainability.

Constraints and Assumptions

This section will outline any design constraints that have been imposed on the design of the system by the customer, thereby removing certain options from being considered by the developers. Also, this section will contain any assumptions that have been made by the requirements engineering team when gathering and analyzing the requirements. If any of the assumptions are found to be false, the system requirements specification would need to be re-evaluated to make sure that the documented requirements are still valid.

Acceptance Criteria

This section will describe the criteria by which the customer will "sign-off" on the final system. Depending on the methodology, this may happen at the end of the testing and quality assurance phase, or in an agile methodology, at the end of each iteration.

The criteria will usually refer to the need to complete all user acceptance tests and the rectification of all defects/bugs that meet a pre-determined priority or severity threshold.

Python Language

Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute your own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems.

Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations.

Python coding style comprises physical lines as well as logical lines or statements. A physical line in a Python program is a sequence of characters, and the end of the line terminates the line sequence as opposed to some other languages, such as C and C++ where a semicolon is used to mark the end of the statement. A logical line, on the other hand, is composed of one or more physical lines. The use of a semi-colon is not prohibited in Python, although itÃ¢â‚¬â„¢s not mandatory. The NEWLINE token denotes the end of the logical line. A logical line that only contains spaces, comments, or tabs are called blank lines and they are ignored by the interpreter.

As we saw that in Python, a new line simply means that a new statement has started. Although, Python does provide a way to split a statement into a multiline statement or to join multiple statements into one logical line. This can be helpful to increase the readability of the statement. Following are the two ways to split a line into two or more lines:

- **Explicit Line Joining**

  In explicit line joining, we use a backward slash to split a statement into a multiline statement.

- **Implicit Line Joining**

  Statements that reside inside [], {}, or () parentheses can be broken down into two or more physical lines without using a back slash.

Multiple Statements on a Single Line

In Python, it is possible to club multiple statements in the same line using a semi-colon; however, most programmers do not consider this to be a good practice as it reduces the readability of the code.

Whitespaces and Indentation

Unlike most of the programming languages, Python uses indentation to mark a block of code. According to Python coding style guideline or PEP8, we should keep an indent size of four.

Most of the programming languages provide indentation for better code formatting and do not enforce to have it. But in Python it is mandatory. This is why indentation is so crucial in Python.

Comments in any programming language are used to increase the readability of the code. Similarly, in Python, when the program starts getting complicated, one of the best ways to maintain the readability of the code is to use Python comments. It is considered a good practice to include documentations and notes in the python syntax since it makes the code way more readable and understandable to other programmers as well, which comes in handy when multiple programmers are simultaneously working on the same project.

Following are different kinds of comments that can be included in our Python program:

- Single Line Comments

  Single line Python comments are marked with # character. These comments end at the end of the physical line, which means that all characters starting after the # character (and lasts till the end of the line) are part of the comment.

- Docstring Comments

  Python has the documentation strings (or docstrings) feature which is usually the first statement included in functions and modules.

  Rather than being ignored by the Python Interpreter like regular comments, docstrings can actually be accessed at the run time using the dot operator.

  It gives programmers an easy way of adding quick notes with every Python module, function, class, and method. To use this feature, we use triple quotes in the beginning of the documentation string or comment and the closing triple quotes at the end of the documentation comment. Docstrings can be one-liners as well as multi-liners.

- Multiline Comments

  Unlike some programming languages that support multiline comments, such as C, Java, and more, there is no specific feature for multiline comments in Python. But that does not mean that it is totally impossible to make multiline comments in Python. There are two ways we can include comments that can span across multiple lines in our Python code.

  Python Block Comments: We can use several single line comments for a whole block. This type of comment is usually created to explain the block of code that follows the Block comment. Python Block comment is the only way of writing a real comment that can span across multiple lines. It is supported and preferred by Pythons PEP8 style guide since Block comments are ignored by Python interpreter or parser.

Data Types

One of the most crucial part of learning any programming language is to understand how data is stored and manipulated in that language. Users are often inclined toward Python because of its ease of use and the number of versatile features it provides. One of those features is dynamic typing.

In Python, unlike statically typed languages like C or Java, there is no need to specifically declare the data type of the variable. In dynamically typed languages such as Python, the interpreter itself predicts the data type of the Python Variable based on the type of value assigned to that variable.

Advantages of Python

- Universal Language Construct
- Support both High Level and Low Level Programming
- Language Interoperability
- Fastest Development life cycle therefore more productive coding environmentLess memory used because a single container hold
- Multiple data types and each type doesnÃ¢â‚¬â„¢t require its own function
- Learning Ease and open source development
- Speed and user-friendly data structure
- Extensive and extensible libraries.
- Simple & support IoT
- and many more

Anaconda Software

Anaconda is the data science platform for data scientists, IT professionals and business leaders of tomorrow. It is a distribution of Python, R, etc. With more than 300 packages for data science, it becomes one of the best platforms for any project.

Anaconda helps in simplified package management and deployment. Anaconda comes with a wide variety of tools to easily collect data from various sources using various machine learning and AI algorithms. It helps in getting an easily manageable environment setup which can deploy any project with the click of a single button.

Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms.

Anaconda Navigator is a desktop GUI that ships with Anaconda and lets you launch applications and manage conda packages, environments, and channels without having to use a command-line interface. It can search for packages in a local Anaconda repository or on Anaconda Cloud. With Navigator, you donâ€™t need to type commands in a terminal, it lets you work with packages and environments with just a click.

Jupyter Notebook

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. In other words: itâ€™s a single document where you can run code, display the output, and also add explanations, formulas, charts, and make your work more transparent, understandable, repeatable, and shareable.

Using Notebooks is now a major part of the data science workflow at companies across the globe. If your goal is to work with data, using a Notebook will speed up your workflow and make it easier to communicate and share your results.

As a server-client application, the Jupyter Notebook App allows you to edit and run your notebooks via a web browser. The application can be executed on a PC without Internet access, or it can be installed on a remote server, where you can access it through the Internet.

Its two main components are the kernels and a dashboard.

- A kernel is a program that runs and introspects the users code. The Jupyter Notebook App has a kernel for Python code, but there are also kernels available for other programming languages.
- The dashboard of the application not only shows you the notebook documents that you have made and can reopen but can also be used to manage the kernels: you can which ones are running and shut them down if necessary.

Jupyter Notebook Features

- Pluggable authentication

  Manage users and authentication with PAM, OAuth or integrate with your own directory service system.

- Centralized deployment

  Deploy the Jupyter Notebook to thousands of users in your organization on centralized infrastructure on- or off-site.

- Container friendly

  Use Docker and Kubernetes to scale your deployment, isolate user processes, and simplify software installation.

- Live coding environments

  Code can be changed and run in real-time with feedback provided directly in the browser

- Code meets data

  Deploy the Notebook next to your data to provide unified software management and data access within your organization.

TensorFlow

Deep learning is a subfield of machine learning that is a set of algorithms that is inspired by the structure and function of the brain. Deep learning is a subset of machine learning. There are certain specialties in which we perform machine learning, and that's why it is called deep learning. For example, deep learning uses neural networks, which are like a simulation of the human brain. Deep learning also involves analyzing large amounts of unstructured data, unlike traditional machine learning, which typically uses structured data. This unstructured data could be fed in the form of images, video, audio, text, etc.

TensorFlow is the second machine learning framework that Google created and used to design, build, and train deep learning models. You can use the TensorFlow library do to numerical computations, which in itself does not seem all too special, but these computations are done with data flow graphs. In these graphs, nodes represent mathematical operations, while the edges represent the data, which usually are multidimensional data arrays or tensors, that are communicated between these edges.

The name TensorFlow is derived from the operations which neural networks perform on multidimensional data arrays or tensors! Its literally a flow of tensors.

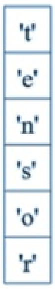
Using tf.keras allows you to design, fit, evaluate, and use deep learning models to make predictions in just a few lines of code. It makes common deep learning tasks, such as classification and regression predictive modeling

The other important aspect is TensorFlow is highly scalable. You can write your code and then make it run either on CPU, GPU, or across a cluster of these systems for the training purpose.

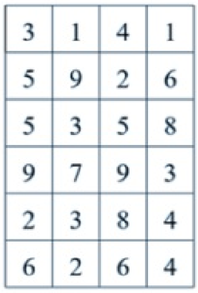
Generally, training the model is where a large part of the computation goes. Also, the process of training is repeated multiple times to solve any issues that may arise. This process leads to the consumption of more power, and therefore, you need a distributed computing. If you need to process large amounts of data, TensorFlow makes it easy by running the code in a distributed manner.

GPUs, or graphical processing units, have become very popular. Nvidia is one of the leaders in this space. It is good at performing mathematical computations, such as matrix multiplication, and plays a significant role in deep learning. TensorFlow also has integration with C++ and Python API, making development much faster.

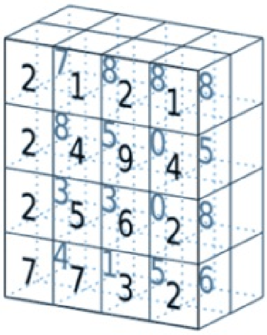
A tensor is a mathematical object represented as arrays of higher dimensions. These arrays of data with different sizes and ranks get fed as input to the neural network. These are the tensors.



Tensor of dimension[1]



Tensor of dimensions[2]



Tensor of dimensions[3]

You can have arrays or vectors, which are one-dimensional, or matrices, which are two-dimensional. But tensors can be more than three, four or five-dimensional. Therefore, it helps in keeping the data very tight in one place and then performing all the analysis around that.

Hardware & Software Requirements

Hardware Requirements

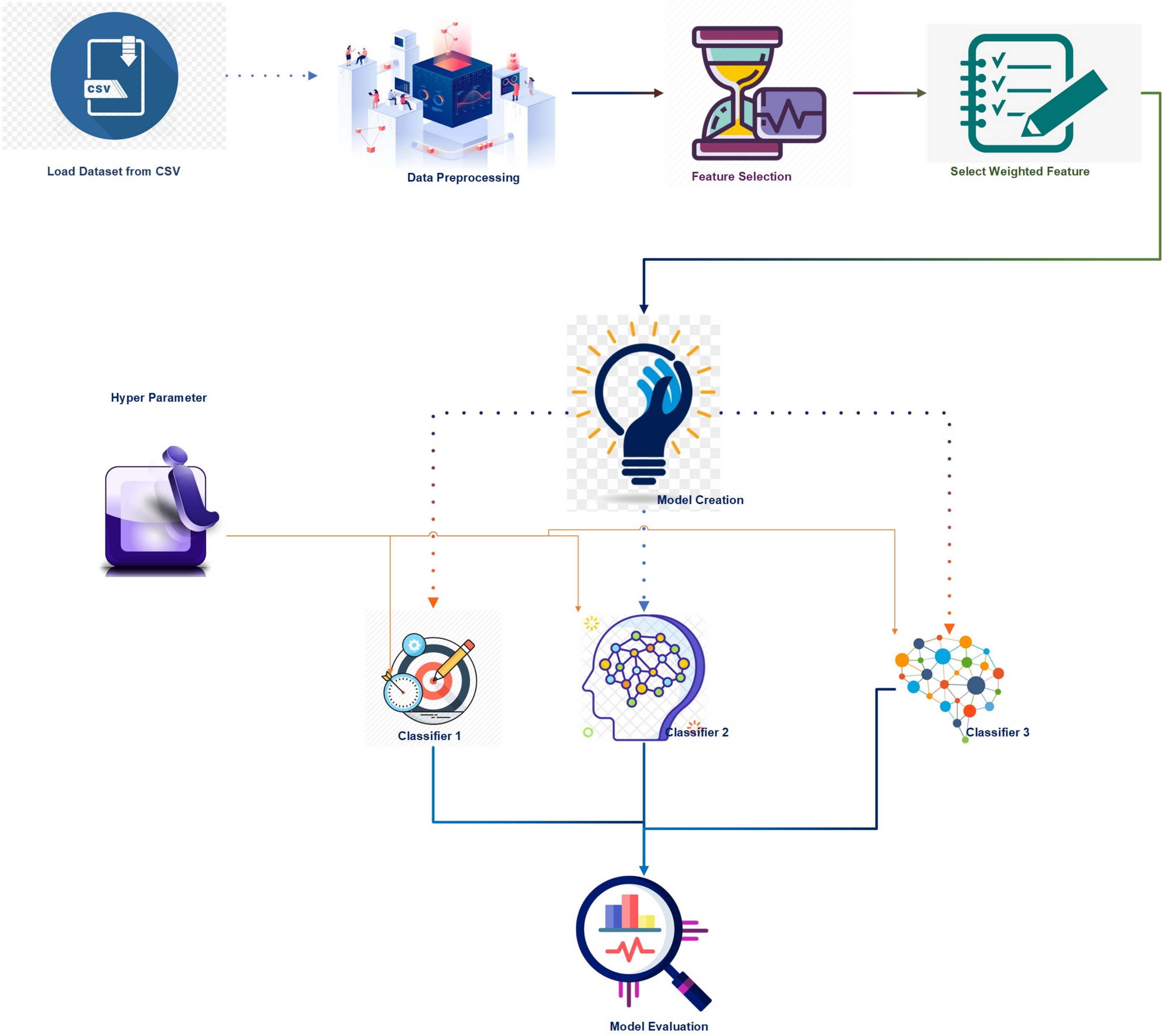
- Processor: Minimum i3 Dual Core
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 100 GB; Recommended 200 GB or more
- Memory (RAM): Minimum 8 GB; Recommended 32 GB or above

Software Requirements

- Python
- Anaconda
- Jupyter Notebook
- TensorFlow
- Keras

Chapter 5: System Design

Architecture



Existing Algorithm

Machine Learning Algorithm

Proposed Algorithm

Long Short Term Memory (LSTM) Algorithm

Advantages of Proposed Algorithm

Long Short Term Memory (LSTM) Algorithm Advantages Advantages

- 👍 **Can be used when dealing with large sequences and accuracy is concerned.**
- 👍 **Improved method of back propagating the error.**
- 👍 **Explicitly designed to deal with the long-term dependency problem.**

Chapter 5: System Implementation

Project Modules

Module 1 : Exploratory Data Analysis

EDA is associated with graphical visualization techniques to identify data patterns and comparative data analysis. EDA is a preferred technique for feature engineering and feature selection processes for data science projects. Some of the widely used EDA techniques are univariate analysis, bivariate analysis, multivariate analysis, bar chart, box plot, pie carat, line graph, frequency table, histogram, and scatter plots. We use EDA to quickly identify any errors in the data. In the Univariate Analysis, we perform data analysis on a single variable. In the Multivariate Analysis, we perform comparative analysis between multiple variables. For any machine learning and deep learning projects, finding data correlations using visual representations is key to identifying dataset insights. Therefore, we explores these insights to set the right path to achieving the accurate model prediction goals.

Module 2 : Feature Selection

In our study of relevant features we employ three standard selection methods: correlation-based univariate, MI-based univariate, and correlationbased forward search algorithms. Correlation (MI) based univariate methods simply rank features based on their correlation (MI) with the target variable. Then the desired number of features is selected based on the ranking. Forward search correlation-based method iteratively selects features based on maximum relevance and minimum redundancy. The relevance is calculated based on the correlation between the feature and the target variable while redundancy is calculated based on the correlation between the feature and the previously selected subset of features.

Module 3 : Model Training and Evaluation

Splitting Datasets: A key characteristic of a good learning model is its ability to generalise to new, or unseen, data. A model which is too close to a particular set of data is described as overfit, and therefore, will not perform well with unseen data. A generalised model requires exposure to multiple variations of input samples. Primarily, models require two sets of data, one to train and another to test. The training data is the set of instances that the model trains on, while the testing data is used to evaluate the generalisability of the model, that is, the performance of the model with unseen data. The train/test split can yield good results; however, this approach has some drawbacks. Although splitting is random, it can happen that the split creates imbalance between the training and the testing set, where the training set has a large number of instances from only one class. In such cases, the model fails to generalise and overfits.

To mitigate this, the datasets are split into three subsets; training, validation and testing. This split is done in a 60:20:20 ratio, for training, validation and testing respectively. The train_test_split helper method from the scikit-learn library is used for the split. With this approach, training is done in two phases, with the training and the validation sets. Firstly, the training set is used to train the model. Then, the validation set is used to estimate the performance of the model on unseen data (data that the model is not trained on).

Parameter Tuning: A parameter can be loosely described as a pre-defined attribute of the data. A parametric algorithm possesses a fixed number of parameters. While a parametric algorithm is computationally more efficient, it makes stronger assumptions about the dataset. This would be ideal if the assumptions are correct. However, parametric algorithms perform poorly with incorrect assumption.

In contrast, non-parametric algorithms are more flexible. In nonparametric scenarios, as the algorithm learns, the

number of parameters grows. This type of algorithm performs slower computations; however, it makes far less assumptions about the dataset.

Chapter 7: Software Testing

Testing documentation is the documentation of artifacts that are created during or before the testing of a software application. Documentation reflects the importance of processes for the customer, individual and organization. Projects which contain all documents have a high level of maturity. Careful documentation can save the time, efforts and wealth of the organization.

If the testing or development team gets software that is not working correctly and developed by someone else, so to find the error, the team will first need a document. Now, if the documents are available then the team will quickly find out the cause of the error by examining documentation. But, if the documents are not available then the tester need to do black box and white box testing again, which will waste the time and money of the organization. More than that, Lack of documentation becomes a problem for acceptance.

Benefits of using Documentation

- Documentation clarifies the quality of methods and objectives.
- It ensures internal coordination when a customer uses software application.
- It ensures clarity about the stability of tasks and performance.
- It provides feedback on preventive tasks.
- It provides feedback for your planning cycle.
- It creates objective evidence for the performance of the quality management system.

The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases.

In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

In the test scenario, testers need to put themselves in the place of the user because they test the software application under the users point of view. Preparation of scenarios is the most critical part, and it is necessary to seek advice or help from customers, stakeholders or developers to prepare the scenario.

As per the IEEE Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report. Hence the testing of all the above mentioned documents is known as documentation testing.

This is one of the most cost effective approaches to testing. If the documentation is not right: there will be major and costly problems. The documentation can be tested in a number of different ways to many different degrees of complexity. These range from running the documents through a spelling and grammar checking device, to manually reviewing the documentation to remove any ambiguity or inconsistency.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed.

The most popular testing documentation files are test reports, plans, and checklists. These documents are used to outline the teams workload and keep track of the process. Lets take a look at the key requirements for these files and see how they contribute to the process.

- Test strategy
  An outline of the full approach to product testing. As the project moves along, developers, designers, product owners can come back to the document and see if the actual performance corresponds to the planned activities.

- Test data
  The data that testers enter into the software to verify certain features and their outputs. Examples of such data can be fake user profiles, statistics, media content, similar to files that would be uploaded by an end-user in a ready solution.

- Test plans
  A file that describes the strategy, resources, environment, limitations, and schedule of the testing process. Its the fullest testing document, essential for informed planning. Such a document is distributed between team members and shared with all stakeholders.

- Test scenarios
  In scenarios, testers break down the productÃ¢â‚¬â„¢s functionality and interface by modules and provide real-time status updates at all testing

stages. A module can be described by a single statement, or require hundreds of statuses, depending on its size and scope.

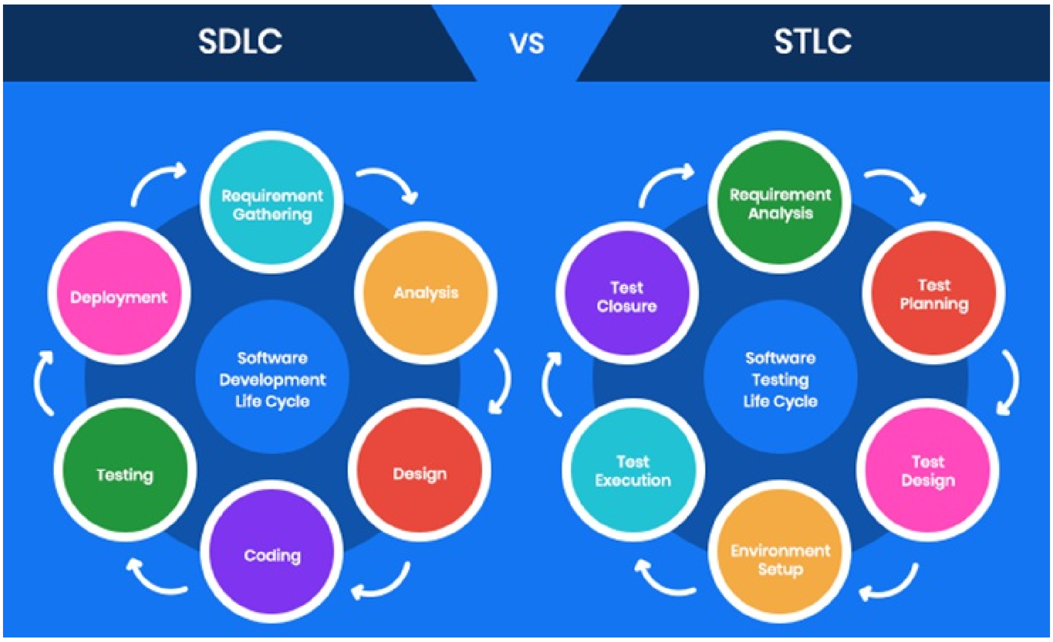
- Test cases
  If the test scenario describes the object of testing (what), a scenario describes a procedure (how). These files cover step-by-step guidance, detailed conditions, and current inputs of a testing task. Test cases have their own kinds that depend on the type of testing, functional, UI, physical, logical cases, etc. Test cases compare available resources and current conditions with desired outcomes and determine if the functionality can be released or not.

- Traceability Matrix
  This software testing documentation maps test cases and their requirements. All entries have their custom IDs Ã¢â‚¬â€ team members and stakeholders can track the progress of any tasks by simply entering its ID to the search.

The combination of internal and external documentation is the key to a deep understanding of all testing processes. Although stakeholders typically have access to the majority of documentation, they mostly work with external files, since they are more concise and tackle tangible issues and results. Internal files, on the other hand, are used by team members to optimize the testing process.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

Software Testing Life Cycle (Taken from Google.com)



Introduction

Unit Testing frameworks are mostly used to help write unit tests quickly and easily. Most of the programming languages do not support unit testing with the inbuilt compiler. Third-party open source and commercial tools can be used to make unit testing even more fun.

List of popular Unit Testing tools for different programming languages:

- Java framework - JUnit
- PHP framework - PHPUnit
- C++ frameworks - UnitTest++ and Google C++
- .NET framework - NUnit
- Python framework - py.test

Software Testing Word Cloud (Taken from Google.com)

Functional Testing is a type of black box testing whereby each part of the system is tested against functional specification/requirements. For instance, seek answers to the following questions,

1. Are you able to login to a system after entering correct credentials?
2. Does your payment gateway prompt an error message when you enter incorrect card number?
3. Does your Add a customer screen adds a customer to your records successfully?

Test Driven Development

Test Driven Development, or TDD, is a code design technique where the programmer writes a test before any production code, and then writes the code that will make that test pass. The idea is that with a tiny bit of assurance from that initial test, the programmer can feel free to refactor and refactor some more to get the cleanest code they know how to write. The idea is simple, but like most simple things, the execution is hard. TDD requires a completely different mind set from what most people are used to and the tenacity to deal with a learning curve that may slow you down at first.

Functional Testing types include:

- Unit Testing
- Integration Testing
- System Testing
- Sanity Testing
- Smoke Testing
- Interface Testing
- Regression Testing
- Beta/Acceptance Testing

Non-functional Testing types include

- Load Testing
- Stress Testing
- Volume Testing
- Security Testing
- Compatibility Testing
- Install Testing
- Recovery Testing
- Reliability Testing
- Usability Testing
- Compliance Testing
- Localization Testing

Unit Testing

UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

A unit can be almost anything you want it to be -- a line of code, a method, or a class. Generally though, smaller is better. Smaller tests give you a much more granular view of how your code is performing. There is also the practical aspect that when you test very small units, your tests can be run fast; like a thousand tests in a second fast.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

Most of us might know the classic definition of Unit Testing

Unit Testing is the method of verifying the smallest piece of testable code against its purpose

If the purpose or requirement failed then the unit test has failed. In simple words, Unit Testing means - writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.

Blackbox Testing

During functional testing, testers verify the app features against the user specifications. This is completely different from testing done by developers which is unit testing. It checks whether the code works as expected. Because unit testing focuses on the internal structure of the code, it is called the white box testing. On the other hand, functional testing checks appÃ¢â‚¬â„¢s functionalities without looking at the internal structure of the code, hence it is called black box testing. Despite how flawless the various individual code components may be, it is essential to check that the app is functioning as expected, when all components are combined. Here you can find a detailed comparison between functional testing vs unit testing.

Integration Testing

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

Component integration testing Testing performed to expose defects in the interfaces and interaction between integrated components. System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Integration tests determine if independently developed units of software work correctly when they are connected to each other. The term has become blurred even by the diffuse standards of the software industry, so I've been wary of using it in my writing. In particular, many people assume integration tests are necessarily broad in scope, while they can be more effectively done with a narrower scope.

As often with these things, it's best to start with a bit of history. When I first learned about integration testing, it was in the 1980's and the waterfall was the dominant influence of software development thinking. In a larger project, we would have a design phase that would specify the interface and behavior of the various modules in the system. Modules would then be assigned to developers to program. It was not unusual for one programmer to be responsible for a single module, but this would be big enough that it could take months to build it. All this work was done in isolation, and when the programmer believed it was finished they would hand it over to QA for testing.

Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. Integration testing is a key aspect of software testing.

System Testing

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the systems compliance with the specified requirements. System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not.

System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product. System testing is a method of monitoring and assessing the behaviour of the complete and fully-integrated software product or system, on the basis of pre-decided specifications and functional requirements. It is a solution to the question "whether the complete system functions in accordance to its pre-defined requirements?"

It's comes under black box testing i.e. only external working features of the software are evaluated during this testing. It does not requires any internal knowledge of the coding, programming, design, etc., and is completely based on users-perspective.

A black box testing type, system testing is the first testing technique that carries out the task of testing a software product as a whole. This System testing tests the integrated system and validates whether it meets the specified requirements of the client.

System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the functional specification or system specification documentation. In most cases, it is done next to the Integration testing, as this testing should be covering the end-to-end systems actual routine. This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system as a whole. This testing checks complete end-to-end scenario as per the customerÃ¢â,¬â„¢s point of view. Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing;

Sanity Testing

Sanity Testing is done when as a QA we do not have sufficient time to run all the test cases, be it Functional Testing, UI, OS or Browser Testing. Sanity testing is a subset of regression testing. After receiving the software build, sanity testing is performed to ensure that the code changes introduced are working as expected. This testing is a checkpoint to determine if testing for the build can proceed or not. The main purpose of this testing is to determine that the changes or the proposed functionality are working as expected. If the sanity test fails, the build is rejected by the testing team to save time and money. It is performed only after the build has cleared the smoke test and been accepted by the Quality Assurance team for further testing. The focus of the team during this testing process is to validate the functionality of the application and not detailed testing.

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as Ã¢â,¬oeDay 0Ã¢â,¬ check. It is done at the build level.

It helps not to waste the testing time to simply testing the whole application when the key features donÃ¢â,¬â„¢t work or the key bugs have not been fixed yet. Here our focus will be on primary and core application work flow.

To conduct smoke testing, we do not write test cases. We just pick the necessary test cases from already written test cases. As mentioned earlier, here in Smoke Testing, our main focus will be on core application work flow. So we pick the test cases from our test suite which cover major functionality of the application. In general, we pick minimal number of test cases that wont take more than half an hour to execute.

The main aim of Sanity testing to check the planned functionality is working as expected. Instead of doing whole regression testing the Sanity testing is perform.

Sanity tests helps to avoid wasting time and cost involved in testing if the build is failed. Tester should reject the build upon build failure. After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software. Typically this is done nearing end of SDLC i.e. while releasing the software. You can say that sanity testing is a subset of acceptance testing. We can also say Tester Acceptance Testing for Sanity testing.

Regression Testing

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

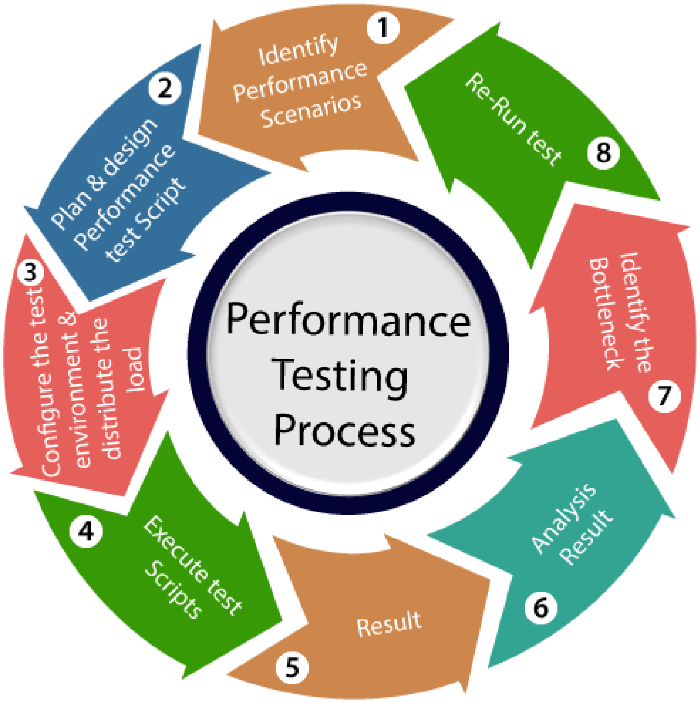
Regression Testing is a Software Testing type in which test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix. For regression testing to be effective, it needs to be seen as one part of a comprehensive testing methodology that is cost-effective and efficient while still incorporating enough varietyÃ¢â‚¬â€such as well-designed frontend UI automated tests alongside targeted unit testing, based on smart risk prioritizationÃ¢â‚¬â€to prevent any aspects of your software applications from going unchecked. These days, many Agile work environments employing workflow practices such as XP (Extreme Programming), RUP (Rational Unified Process), or Scrum appreciate regression testing as an essential aspect of a dynamic, iterative development and deployment schedule. But no matter what software development and quality-assurance process your organization uses, if you take the time to put in enough careful planning up front, crafting a clear and diverse testing strategy with automated regression testing at its core, you can help prevent projects from going over budget, keep your team on track, and, most importantly, prevent unexpected bugs from damaging your products and your companys bottom line.

Performance testing

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size.

Performance Testing (Taken from Google.com)


Performance Testing Process
1 Identify Performance Scenarios
2 Plan & design Performance test Script
3 Configure the test environment & distribute the load
4 Execute test Scripts
5 Result
6 Analysis Result
7 Identify the Bottleneck
8 Re-Run test

Performance testing gathers all the tests that verify an applications speed, robustness, reliability, and correct sizing. It examines several indicators such as a browser, page and network response times, server query processing time, number of acceptable concurrent users architected, CPU memory consumption, and number/type of errors which may be encountered when using an application. Performance testing is the testing that is performed to ascertain how the components of a system are performing under a certain given situation. Resource usage, scalability, and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of a software product.

Software Performance testing is type of testing perform to determine the performance of system to major the measure, validate or verify quality attributes of the system like responsiveness, Speed, Scalability, Stability under variety of load conditions. The system is tested under a mixture of load conditions and check the time required responding by the system under varying workloads. Software performance testing involves the testing of application under test to ensure that application is working as expected under variety of load conditions. The goal of performance testing is not only find the bugs in the system but also eliminate the performance bottlenecks from the system.

Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value. Here Increasing load means increasing number of concurrent users, transactions & check the behavior of application under test. It is normally carried out underneath controlled environment in order to distinguish between two different systems. It is also called as Ã¢â,¬oeEndurance testingÃ¢â,¬ and Ã¢â,¬oeVolume testingÃ¢â,¬. The main purpose of load testing is to monitor the response time and staying power of application when system is performing well under heavy load. Load testing comes under the Non Functional Testing & it is designed to test the non-functional requirements of a software application.

Load testing is perform to make sure that what amount of load can be withstand the application under test. The successfully executed load testing is only if the specified test cases are executed without any error in allocated time.v

Testing printer by sending large job. Editing a very large document for testing of word processor Continuously reading and writing data into hard disk. Running multiple applications simultaneously on server. Testing of mail server by accessing thousands of mailboxes In case of zero-volume testing & system fed with zero load.

Chapter 8: Conclusion

While the absence of datasets was the very focal point at which this study was conducted, it can also be seen as a limitation on its own given the fact that potentially more accurate results would have been obtained on the comparison between the datasets.

Our attack detection method for public peering points has enabled us to unveil distributed inter-domain attacks. Our results show that the DNS attack vector is more popular than previously captured by (even distributed) honeypots, a common vantage point in the context of reflection and amplification attacks. We were successful in tracking a prominent attack entity and identifying concrete attack patterns. Our study reveals that attackers are able to detect new abusable amplifiers quickly and reasonably change which infrastructure they abuse. At the same time, we find that attackers could achieve higher amplification by choosing (query) names more prudently. especially in the case of attacks utilizing spoofing and highly variable amplifier sets.

There were a number of time-related features that we did not capture in our clustering, but found particularly compelling. Some generators appear to use a fixed number of labels at any time, which changes over the attack, presumably from building attack queries by continually appending, or removing, labels. In this particular example, the number of labels descends in time, but in many other examples it is seen to ascend. This would be consistent with the use of a dictionary, which is used to select a label, and creating queries by continually appending labels.

Chapter 9: Future Work

There are numerous avenues for further research into this area the results of which can help us understand cyber actors better and may lead to techniques that can be applied to a broader set of problems. In our research, we did not study the victims, for example, which may further refine our understanding of both the generators and the actors operating these attacks. One could take a number of graph approaches to these problems, including a study of how the labels form tightly connected clusters among attacks.

Chapter 10: Appendix - I Screenshots

Chapter 11: Appendix - II Sample Coding

Chapter 12: References

- Muhammad Sulaiman,Muhammad Waseem,Addisu Negash Ali,Ghaylen Laouini,Fahad Sameer Alshammari Defense Strategies for Epidemic Cyber Security Threats: Modeling and Analysis by Using a Machine Learning Approach IEEE Access, 2024
- Jichao Bi,Fangfei Zhang,Ali Dorri,Chunming Zhang,Chen Zhang A Risk Management Approach to Double-Virus Tradeoff Problem IEEE Access, 2019
- Abdul Basit Ajmal,Munam Ali Shah,Carsten Maple,Muhammad Nabeel Asghar,Saif Ul Islam Offensive Security: Towards Proactive Threat Hunting via Adversary Emulation IEEE Access, 2021
- Abel Yeboah-Ofori,Shareeful Islam,Sin Wee Lee,Zia Ush Shamszaman,Khan Muhammad,Meteb Altaf,Mabrook S. Al-Rakhami Cyber Threat Predictive Analytics for Improving Cyber Supply Chain Security IEEE Access, 2021
- Ana Kovacevic,Nenad Putnik,Oliver ToÅ¡kovic Factors Related to Cyber Security Behavior IEEE Access, 2020
- Nan Sun,Chang-Tsun Li,Hin Chan,Md Zahidul Islam,Md Rafiqul Islam,Warren Armstrong How Do Organizations Seek Cyber Assurance? Investigations on the Adoption of the Common Criteria and Beyond IEEE Access, 2022
- Haichun Zhang,Yuqian Pan,Zhaojun Lu,Jie Wang,Zhenglin Liu A Cyber Security Evaluation Framework for In-Vehicle Electrical Control Units IEEE Access, 2021
- Alladean Chidukwani,Sebastian Zander,Polychronis Koutsakis A Survey on the Cyber Security of Small-to-Medium Businesses: Challenges, Research Focus and Recommendations IEEE Access, 2022
- Abdul Basit Ajmal,Masoom Alam,Awais Abdul Khaliq,Shawal Khan,Zakria Qadir,M. A. Parvez Mahmud Last Line of Defense: Reliability Through Inducing Cyber Threat Hunting With Deception in SCADA Networks IEEE Access, 2021
- Abdul Wahid Khan,Shah Zaib,Faheem Khan,Ilhan Tarimer,Jung Taek Seo,Jiho Shin Analyzing and Evaluating Critical Cyber Security Challenges Faced by Vendor Organizations in Software Development: SLR Based Approach IEEE Access, 2022
- O. David, S. Sarkar, N. Kammerer, C. Nantermoz, F. M. de Chamisso, B. Meden, et al., Digital assistances in remote operations for ITER test blanket system replacement: An experimental validation, Fusion Eng. Des., vol. 188, Mar. 2023.
- P. Xiao, Z. Qin, D. Chen, N. Zhang, Y. Ding, F. Deng, et al., FastNet: A lightweight convolutional neural network for tumors fast identification in mobile-computer-assisted devices, IEEE Internet Things J., vol. 10, no. 11, pp. 9878-9891, Jun. 2023.
- A. S. Alsafran, A feasibility study of implementing IEEE 1547 and IEEE 2030 standards for microgrid in the kingdom of Saudi Arabia, Energies, vol. 16, no. 4, pp. 1777, Feb. 2023.
- R. Pinciroli and C. Trubiani, Performance analysis of fault-tolerant multi-agent coordination mechanisms, IEEE Trans. Ind. Informat., vol. 19, no. 9, pp. 9821-9832, Sep. 2023.
- M. Aizat, A. Azmin and W. Rahiman, A survey on navigation approaches for automated guided vehicle robots in dynamic surrounding, IEEE Access, vol. 11, pp. 33934-33955, 2023.
- M. Jalili and M. Perc, Information cascades in complex networks, J. Complex Netw., vol. 5, pp. 665-693, 2017.
- P. SzÃµr, The Art of Computer Virus Research and Defense, Hagerstown, MD, USA:Pearson, 2005.
- M. H. R. Khouzani, S. Sarkar and E. Altman, Optimal dissemination of security patches in mobile wireless

networks, IEEE Trans. Inf. Theory, vol. 58, no. 7, pp. 4714-4732, Jul. 2012.

» S. Eshghi, M. H. R. Khouzani, S. Sarkar and S. S. Venkatesh, Optimal patching in clustered malware epidemics, IEEE/ACM Trans. Netw., vol. 24, no. 1, pp. 283-298, Feb. 2014.

» C. Nowzari, V. M. Preciado and G. J. Pappas, Analysis and control of epidemics: A survey of spreading processes on complex networks, IEEE Control Syst., vol. 36, no. 1, pp. 26-46, Feb. 2016.

» H. Lin and N. W. Bergmann, IoT privacy and security challenges for smart home environments, Information, vol. 7, no. 3, pp. 44, 2016.

» N. M. Karie, N. M. Sahri and P. Haskell-Dowland, IoT threat detection advances challenges and future directions, Proc. IEEE Workshop Emerg. Technol. Secur. IoT (ETSecIoT), pp. 22-29, Apr. 2020.

» V. R. Kebande, N. M. Karie and H. S. Venter, Adding digital forensic readiness as a security component to the IoT domain, Int. J. Adv. Sci. Eng. Inf. Technol., vol. 8, no. 1, pp. 1-11, 2018.

» W. M. S. Stout and V. E. Urias, Challenges to securing the Internet of Things, Proc. IEEE Int. Carnahan Conf. Secur. Technol. (ICCST), pp. 1-8, Oct. 2016.

» Z. A. Solangi, Y. A. Solangi, S. Chandio, M. B. S. A. Aziz, M. S. B. Hamzah and A. Shah, The future of data privacy and security concerns in Internet of Things, Proc. IEEE Int. Conf. Innov. Res. Develop. (ICIRD), pp. 1-4, May 2018.

» P. S. Shinde and S. B. Ardhapurkar, Cyber security analysis using vulnerability assessment and penetration testing, Proc. World Conf. Futuristic Trends Res. Innov. Social Welfare (Startup Conclave), pp. 1-5, Feb. 2016.

» P. Dholey and A. K. Shaw, OnlineKALI: Online vulnerability scanner, Proc. Int. Ethical Hacking Conf. Adv. Intell. Syst. Comput., vol. 811, pp. 25-35, 2019.

» P. Russo, A. Caponi, M. Leuti and G. Bianchi, A web platform for integrated vulnerability assessment and cyber risk management, Information, vol. 10, no. 7, pp. 242, Jul. 2019.

» R. Stevens, D. Votipka, E. M. Redmiles, C. Ahern and M. L. Mazurek, Applied digital threat modeling: It works, IEEE Secur. Privacy, vol. 17, no. 4, pp. 35-42, Jul. 2019.

» J. M. Archibald and K. Renaud, Refining the PoinTER â€˜human firewallâ€™ pentesting framework, Inf. Comput. Secur., vol. 26, no. 4, pp. 575-600, 2019.

» B. Woods and A. Bochman, Supply chain in the software era in Scowcroft Center for Strategic and Security, Washington, DC, USA:Atlantic Council, May 2018.

» A. Yeboah-Ofori and F. Katsriku, Cybercrime and risks for cyber physical systems, Int. J. Cyber-Secur. Digit. Forensics, vol. 8, no. 1, pp. 43-57, 2019.

» R. D. Labati, A. Genovese, V. Piuri and F. Scotti, Towards the prediction of renewable energy unbalance in smart grids, Proc. IEEE 4th Int. Forum Res. Technol. Soc. Ind. (RTSI), pp. 1-5, Sep. 2018.

» J. Boyens, C. Paulsen, R. Moorthy and N. Bartol, Supply chain risk management practices for federal information systems and organizations, NIST Comput. Sec., vol. 800, no. 161, pp. 32, 2015.

» Framework for Improving Critical Infrastructure Cybersecurity Version 1.1, Gaithersburg, MD, USA, 2018.

» F.-J. Hinojo-Lucena, I. Aznar-Diaz, M.-P. Caceres-Reche, J.-M. Trujillo-Torres and J.-M. Romero-Rodriguez, Factors influencing the development of digital competence in teachers: Analysis of the teaching staff of permanent education centres, IEEE Access, vol. 7, pp. 178744-178752, 2019.

» K. Parsons, D. Calic, M. Pattinson, M. Butavicius, A. McCormac and T. Zwaans, The human aspects of information security questionnaire (HAIS-Q): Two further validation studies, Comput. Secur., vol. 66, pp. 40-51, May 2017.

» B. D. Sawyer and P. A. Hancock, Hacking the human: The prevalence paradox in cybersecurity, Hum. Factors J.

Hum. Factors Ergonom. Soc., vol. 60, pp. 597-609, Aug. 2018.

- B. K. Wiederhold, The role of psychology in enhancing cybersecurity, Cyberpsychol. Behav. Social Netw., vol. 17, no. 3, pp. 131-132, Mar. 2014.
- D. E. de Zafra, S. I. Pitcher, J. D. Tressler, J. B. Ippolito and M. Wilson, Information technology security training requirements?: A role- and performance-based model, 1998.
- N. Sun, J. Zhang, P. Rimba, S. Gao, Y. Xiang and L. Y. Zhang, Data-driven cybersecurity incident prediction: A survey, IEEE Commun. Surveys Tuts., vol. 21, no. 2, pp. 1744-1772, 2nd Quart. 2018.
- S. N. Matheu, J. L. HernÃ¡ndez-Ramos, A. F. Skarmeta and G. Baldini, A survey of cybersecurity certification for the Internet of Things, ACM Comput. Surv., vol. 53, no. 6, pp. 1-36, Nov. 2021.
- D. S. Herrmann, Using the Common Criteria for IT Security Evaluation, Boca Raton, FL, USA:CRC Press, 2002.
- N. Sun, C.-T. Li, H. Chan, B. D. Le, M. Islam, L. Y. Zhang, et al., Defining security requirements with the common criteria: Applications adoptions and challenges, IEEE Access, vol. 10, pp. 44756-44777, 2022.
- W. Stallings, L. Brown, M. D. Bauer and A. K. Bhattacharjee, Computer Security: Principles and Practice, Upper Saddle River, NJ, USA:Pearson, 2012.
- K. B. Kelarestaghi, M. Foruhandeh, K. Heaslip and R. Gerdes, Intelligent transportation system security: Impact-oriented risk assessment of in-vehicle networks, IEEE Intell. Transp. Syst. Mag., vol. 13, no. 2, pp. 91-104, May 2021.
- T. T. Dandala, V. Krishnamurthy and R. Alwan, Internet of vehicles (IoV) for traffic management, Proc. Int. Conf. Comput. Commun. Signal Process. (ICCCSP), pp. 1-4, Jan. 2017.
- W. Yanbang, Y. Jing and Y. Zhilou, Auto-driving vehicle testing method ECU and system, 2018.
- H. Pingguo, Y. Jingjing and C. Xiao, Security access control method for vehicle diagnosis system, Jun. 2014.
- R. Q. Malik, K. N. Ramli, Z. H. Kareem, M. I. Habelalmatee, A. H. Abbas and A. Alamoody, An overview on V2P communication system: Architecture and application, Proc. 3rd Int. Conf. Eng. Technol. Appl. (IICETA), pp. 174-178, Sep. 2020.
- A. Vives, Social and environmental responsibility in small and medium enterprises in Latin America, J. Corporate Citizenship, vol. 2006, no. 21, pp. 39-50, Mar. 2006.
- K. Renaud and G. R. S. Weir, Cybersecurity and the unbearability of uncertainty, Proc. Cybersecurity Cyberforensics Conf. (CCC), pp. 137-143, Aug. 2016.
- K. Renaud and G. R. S. Weir, Cybersecurity and the unbearability of uncertainty, Proc. Cybersecurity Cyberforensics Conf. (CCC), pp. 137-143, Aug. 2016.