Chapter One

Introduction

1.1 Background to the Study

Information Technology is becoming a significant part of our daily life, and one used to a great extent, to help solve difficulties since we live in a dynamic and fast changing environment. Web-based systems and applications have always been a significant part of this Information Technology[1]. Generally, Information Technology is becoming more crucial now that many of critical infrastructures, i.e. health sector, banking sector, business sector, commutations sector, and networking are depending on them. The World Wide Web is expanding daily and the internet has become a necessity for everyone. The use of the internet by individuals, academia, government, business and organizations across a variety of industries, has drastically expanded during the past decade[1]. Internet information resources are rapidly expanding and are now present in many aspects of daily life. Web applications can provide excellent digital experiences, but only those that are secure can properly deliver this services[2].

Web-based systems provide organization and businesses with quick and easy operation through digitalization and automation of process. However, web-based applications vulnerabilities in codding and databases can now be exploited by attackers to gain illegal access to user system and personal information[2]. Attackers now target web applications explicitly and create variety of malicious web content in an effort to obtain information from a specific company or individual who is in possession of valuable information[3].

Even though most of these organizations install firewall, antimalware, and other traditional security systems, they are still unable to stop the vast majority of Web attacks. The Identification and prevention of various forms of cyberattacks are becoming more difficult due to the limits of traditional security methods. It is getting harder to secure the integrity confidentiality and availability of web-based systems even though users rely more and more on them. Due to the sensitive, valuable, and sufficient information collected from users and held by these systems, web-based systems and applications are frequently the target of hackers who want to steal information, make money, or engage in other illegal actions, leading to disastrous effects on the finances and reputation of system users[3]. Web-based attacks are transmitted as internet and web-based system usage grows. Between 2020 and 2021, the number of fraudulent web application requests increased by 88% with more than 75% been injection and broken access control attacks. Banking and finance sector, along with Software as a Service providers, were the most frequently attacked sectors in 2021, combining for more than 28% of web-based attacks.

Information-Technology Promotion Agency (IPA) estimates that more than 75% of attacks are now targeted against web applications[4]. And Over 80% of online applications on the internet have at least one significant vulnerability[5]. According to the 2017 Internet Security Treat Report (ISTR) more than 76% of the websites that were scanned were deemed to be vulnerable[4].

This vulnerability is of various categories, however the top know web-based attacks mostly employ SQL injection and Cross-site scripting. SQL Injection is a type of web attack that target database-driven website and involve an intruder inserting malicious SQL queries into

the database using data sent from the client to the server[4]. These attacks have grown as more web applications are made available in the cloud, posing a serious danger to web-based services and various web application programs. Cross-site scripting is a form of attack in which the victims' web browser is used to run or download malicious script from remote online pages[5].

Due to the increase in web-based applications and systems and the inability of traditional system to defend against this attack efficiently, attackers are now focusing more on exploiting web-based vulnerability. However, machine learning technique which has been employed by several researcher/study in the field of Cybersecurity and Data mining for the task of predicting attack before they occur based on knowledge acquired from data[6]. This approach can also be used against web-attack by predicting this attack before they actually occur. In terms of web-based attack security, intrusion detection methodology is still relatively new. IDS are generally made to monitor and find intrusive online activity. However, network-based assaults differ significantly from web-based attacks in terms of their properties[7].

Machine learning predictive models learn by looking for patterns in a set of input data. It uses classification algorithm to classify data and foretell future events. It is an essential part of predictive analytics, a sort of data analytics that makes use of both recent and old data to predict activity, behavior, and trends[5]. Predictive modeling is a statistical method that uses data mining and machine learning to predict and anticipate likely future outcomes using historical and existing data[6].

Attackers carefully target organizations by exploring its vulnerabilities and infiltrating their network and control systems using many different types of attack: Trojan horse, Viruses, Worms, Ransomware, Man-in-the-Middle Attack, Denial-of-Service Attack (DoS), Distributed Denial-of-Service Attack (DDoS), SQL Injection, Cross-Site Scripting, User/Root Access Compromise, Phishing Attacks and Zero-Day-Attack[6]. They are indeed ready to incur great costs, time, and expertise in order to accomplish their goals. The terms cyber-attack, cyber threat, adversaries and intrusion are sometimes used to describe these attacks.

Cyber threat refers to human resources who attempt to illegally access network, regulate system and steal information, through the aid of a data communication pathway. It has the capability of damaging and gaining unauthorized access to computers, computer networks and information system[7]. Cyber-attack is any form of injury done to the computer system, network or individual through the internet, they are intra-computer attacks that undermine the confidentiality, integrity and availability of computer system, computer network and their data's. The complexity and number of cyber threats and attacks have increased drastically in recent years, statistics have shown that organization are witnessing an alarming increase in the number of attacks, and that attack scenarios are also varying[6]. McAfee report estimated that global losses from cybercrime is around 1 trillion USD in 2020, and that they are expected to increase to more than 6 trillion USD in 2021[7].

Every American organization in both public and private sector has been hacked, infected with malware, or be a target of hostile Nation-state cyber intruders, and that there is a business breach every 39 seconds of the day even with the use of traditional cybersecurity approach, "cybercrime is out of control". This implies that traditional and blacklist

approaches like Firewalls, Anti-Malware, User Authentication, Access Control, Cryptography Systems, and Signature-based IDS are unable to detect most of these attacks as the trend is constantly evolving in scale and sophistication. Nevertheless, Anomaly-Based Intrusion Detection System (ABIDS) that use Machine Learning Techniques and Advance Analytics to create systems that can learn from intrusion data and discover knowledge that can be employed to predict same or similar attacks, can be an efficient additional system for attack detection of all kinds, be it external or internal attacks[7].

1.2 Statement of the Problem

Users have faith that the private and secure handling of their sensitive personal information on the website like their credit card, social security, medical information becomes public due to intrusion in the form of Web- based attacks with potentially serious repercussions.

Cyber-intrusion and cyber-security continue to be a serious issue for any sector in the cyberspace as a number of security breaches keep increasing, and modern attacks keep evading traditional cybersecurity procedures and blacklist approach by using impressive highly sophisticated techniques, they can be difficult or even impossible to detect even though most of these attacks are variants of previously known attacks with known signatures. It is known that thousands of zero-day attacks are continuously emerging because of the addition of various protocols, mainly from the field of Internet of Things (IoT). In 2020 Cybersecurity Ventures predict that global cybercrime costs will grow by 15% each year for the next five years, reaching at least 10.5 trillion USD annually by 2025, up from $3 trillion USD in 2015[5]. The United State Government alone invested over 16 billion USD for cyber security and defense in the 2019 fiscal year's budget, which increase to almost 19 billion

USD in 2020 fiscal year's budget[8]. This now leads to many researchers and study, leveraging different data science techniques and artificial intelligence to create anomaly-based IDSs that can detect unknown attacks, using different architecture, methods, approaches, and algorithms such as statistics, data mining, machine learning techniques, advance analytics, and hybridization of system. However, several of these prediction models have suffered from lack of good performance due to the dataset used for training, algorithm used to create the model and/or flaws in the architecture.

Therefore, this study proposes a predictive model for intrusion detection, that uses Data Analytics for feature scaling and feature selection, in order to select the most relevant features and to improve the model performance, and machine learning techniques for classification and prediction of attack.

1.3 Aim and Objectives

The aim of this study is to Comparatively Evaluate the Performance of Random Forest on Web-based Attacks.

The objectives of this study are to:

i. develop a model for the prediction of web-based attacks using random forest.
ii. enhance the performance of the model through feature selection.
iii. evaluate the performance of the developed model before and after feature selection and perform comparative analysis on raw dataset.

1.4 Scope of the Study

This section of the research aids the reader in describing the study's scope and outlining the study's boundaries. The scope of this research center on web-based attacks using feature selection and random forest machine learning algorithm with the dataset of Canadian Institute for Cybersecurity.

1.5 Significance of the Study

The researcher must defend the study's expected contribution to academic research and literature in the subject of study, as well as corporate or operational practice and policy, in the justification of the study. The goal is to design and develop a predictive model for web-based attacks using correlation-based feature selection and random forest algorithm with the dataset of Canadian Institute for Cybersecurity (CIC). The study will add to the body of existing knowledge in the fields of Cybersecurity, web-based attacks, information security, and systems while providing numerous values and benefits to web developers, computer programmers, and computer scientists in light of the development of information technology. The outcome of this study will also serve as a reference material to students of computer science, lecturers and researchers. It can also propel further research on the topic.

1.6 Limitation of the Study

The project is limited to the development of a predictive model for web-based attacks using random forest and feature selection and comparative performance evaluation of random forest with or without feature selection. Also, this research work limited to the web attacks of cross – site scripting (xss), brute force and sql injection.

1.7 Operational Definition of Terms

Web-based: Web-based refers to a deployment technique that requires zero installation and uses the HTTP or HTTPS protocol to launch clients anywhere on a network. It is any program that is used to access resources across a network utilizing HTTP as opposed to being stored locally on a device.

Attacks: Attacks refer to any attempt to cause harm to someone through physical force, verbal abuse, or the launch of a military offensive. It is to use force against (someone or something) in an effort to harm, harm, or destroy (someone or something).

Web-based Attacks: Web-based attacks are a tempting way for threat actors to trick their targets by leveraging web systems and services as the danger vector. In order to alter a web application and retrieve the needed information, it is an attack in which certain data is injected into the program.

Model: A model is a tangible illustration of the appearance or operation of something. A system, entity, phenomenon, or process is represented by a physical, mathematical, or other logical model.

Machine Learning: Machine learning is a subfield of artificial intelligence and computer science that focuses on using data and algorithms to mimic the process of human learning while continuously increasing the accuracy of the model. Building systems that learn or enhance performance based on the data they ingest is the focus of machine learning (ML), a subset of artificial intelligence (AI).

Cybercrime: Cybercrime is defined as any unlawful conduct involving a computer, networked device, or a network. Cybercrime refers to any illegal behavior that involves, targets, or otherwise involves a computer, computer network, or networked device.

Cyber Security: Cybersecurity is the technique of preventing harmful assaults on computers, servers, mobile devices, electronic systems, networks, and data. The use of technologies, procedures, and policies to defend systems, networks, software, hardware, and data from online threats is known as cyber security. Cybersecurity is the safeguarding against cyberthreats of internet-connected systems, including data, software, and hardware.

Dataset: A dataset is a grouping of connected, distinct pieces of connected data that can be accessed singly, in combination, or handled as a whole. A data set is arranged into a certain kind of data structure.

Endnotes

[1] A. Singh, A. Kumar & A. K. Bharti, " *Identification and Prevention approaches for Web-based Attacks using Machine Learning Techniques,"* **International Journal of Creative Research Thoughts (IJCRT)** 2, vol. 9, 2021, pp 4558-4563.

[2] R. Chowdhurry, P. Banerjee, S. Deep Dey, B. Saha & S. K. Bandyopadhyay, *"A Decision Tree Based Intrusion Detection System For Identification of Malicious Web Based Attacks,"* **Preprints** (www.preprints.org), vol. 1, 2020.

[3] G. Acar, D. Y. Huang, F. Li, A. Narayanan & N. Feamster *"Web Based Attacks to Discover and Control Local IoT Devices,"* **In IoT S&P: ACM SIGCOMM** 2018, pp 29-35 https://doi.org/10.1145/3229565.3229568

[4] T.S. Riera, J. B. Higuera, J. M. Herraiz & J. S. Montalvo *"A New Multi-label Dataset for Web Attacks CAPEC Classification using Machine Learning Techniques,"* **Journal of Science Direct Computer and Security** 120, 2022, pp 1-18

[5] M. H. U. Sharif, *"Web Attacks Analysis and Mitigation Techniques,"* **International Journal of Engineering Research and Technology (IJERT),** 2022, www.ijert.org

[6] M. Zwilling, G. Klien, D. Lesjak, L. Wiechetek, F. Cetin & H. N. *Basim "CyberSecurity Awareness, Knowledge and Behavior: A Comparative Study,"* **Journal of Computer Information System,** 2020, pp 1-16

[7] N. Agarwal & S. Z. Hussain, *"A Closer Look at Intrusion Detection System for Web Applications,"* **Hindawi Security and Communication Networks,** 2018, pp 27 https://doi.org/10.1155/2018/9601357

[8] B. Karuparthi & A. Mahesh, *"Comparative Study between Random Forest and Support Vector Machine Algorithm in Classifying Cervical Cancer,"* **International Journal of Engineering Research & Technology (IJERT),** Vol. 11 Issue 01, 2022, pp 434-437

[9] W. Li, *"Optimization and Application of Random Forest Algorithm for Applied Mathematics Specialty,"* **Hindawi Security and Communication Networks,** 2022, pp 9 https://doi.org/10.1155/2022/1131994

[10] S. Singh, S. S. Choudhary & S. Bhavishya, *"Feature Selection Effects on Classification Algorithms: Laconic description of Machine Learning Algorithms,"* **International Journal of Engineering Research & Technology (IJERT)**, Vol. 7 Issue 02, 2018, pp 183-185 http://www.ijert.org

[11] L. Al-Shalabi, *"New Feature Selection Algorithm Based on Feature Stability and Correlation,"* **Journal of Information Technology and Computing,** 2017, pp 1-16 DOI 10.1109/ACCESS.2022.3140209, IEEE Access

[12] Z. Avkurova, *"Models for Early Web – Attacks Detection and Intruders Identification Based on Fuzzy Logic,"* **Journal of Science Direct: Procedia Computer Scienec** 198, 2022, pp 694-699.

[13] Z. Avkurova, S. Gnatyuk, B. Abduraimova, S. Fedushko, Y. Syerov & O. Trach, *''Detecting web Attacks using Random Under sampling and Ensemble Learners,"* **Journal of Big Data**, 2021, pp 1-20 https://doi.org/10.1186/s40537-021-00460-8

Chapter Two

Literature Review

2.1 Conceptual Review

Alongside the rise in cyberattacks, the internet and web-based applications have been expanding quickly. Requests that are seen as normal or odd are used to carry out these assaults (attack requests). As a result, an intrusion attempt could be a classification issue. In order to improve the security of web services, machine learning algorithms are employed to train models to categorize this request[1].

Computers play a respectably significant role in our daily lives. Important information is transmitted and received over the internet, which is today extremely vulnerable to attacks. Attacks on websites are regarded as the most crucial factor in compromising network security. Health care, banking, and online commercial operations are just a few examples of web apps. Confidentiality, integrity, and availability are three essential components of security that must be maintained at all times[1].

2.1.1 Predictive Model

Predictive modeling is a commonly used statistical technique to predict future events; these are data mining technology solutions that evaluate both historical and recent data to create a model that predicts future behavior from data.

To forecast untested behavior, predictive models are utilized. Users and developers of predictive models need to be well-versed in data, statistics, business operations of a company, and the market in which it competes. Many different explanatory factors, only some of which

the researcher is directly interested in, are frequently included in predictive models. Data scientists frequently develop predictive models that utilize statistics to forecast outcomes. To estimate the likelihood that a set of data belongs to another set, models may include one or more classifiers[2].

2.1.2 CyberSecurity

Cybersecurity is a rising issue that is connected to everything a person or organization does that is made possible by the Internet. Given the very unpredictable nature of when, how, where, and by whom threats may originate from, it is a high risk situation and must be managed as such[3]."Collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance, and technologies that can be used to protect the cyber environment, organizations, and users' assets," according to the definition of cybersecurity[3].

The sophistication of system protection tools and cyberattacks has increased, leading to an increase in data breaches and other known cyberattacks. Organizational cybersecurity attacks have not stopped and never will. Information is becoming more and more important to firms as they adopt Internet-enabled business solutions and procedures. Web-based attacks are among the most frequent cyberattacks, it is said[3].

Unfortunately, the growing number of cybersecurity-related events that are being reported internationally shows that algorithms, systems, and processes by themselves are unable to keep digital systems secure. For instance, a recent report suggested that cybersecurity-related accidents cost the world economy up to $600 billion USD in just 2018[4].

Information security experts consider cyber-security attacks to be one of the hottest subjects right now. If not managed appropriately, these attacks can cause significant losses for enterprises. Attacks on cybersecurity are typically the result of a number of factors include threats, mistakes made by humans, or a lack of information[7].Technology, policies, procedures, and information assets are all related to cybersecurity, which aims to prevent any harm or illegal access brought on by cyberattacks. Particularly cyberattacks on information systems have a direct impact on the business-supporting operational procedures, potentially resulting in corporate stagnation[7].

2.1.3 Web-Based Attacks

An organization's network can be attacked by hackers through unprotected Web applications. According to statistics, 42% of web apps are vulnerable to threats and hackers. Although the widespread use of web-based apps and the emphasis on data storage on the internet have been productive and beneficial, they have also made the system's flaws more obvious[1].

Due to the ongoing increase in web threats, web application security is currently one of the most important challenges in information security. Over 76% of the websites scanned were determined to be vulnerable, according to the Internet Security Treat Report (ISTR) 2017. Additionally, according to the research, there were 35% more web-based security breaches in the first quarter of 2017 than there were in the same period in 2016[5].

The Internet plays a significant role as a life-supporting infrastructure in today's information society. Many users utilize the Internet to access a variety of services, including e-mail, weblogs, social networking services (SNSes), and e-commerce sites. It is used by businesses and organizations to deliver and enhance their services. The innovations brought about by the

Internet have an impact on social systems, such as financial and transportation networks, in addition to utilities like gas and water, and significantly increase our daily convenience[2].

On the other hand, as the information society develops, cyberattacks are growing in frequency. Attackers use the Internet to gain unauthorized access to clients and servers owned by others and carry out data leakage, defacement, and destruction. For instance, hackers force businesses into bankruptcy by leaking sensitive information and stealing privacy information from an endless number of customers. Cyberattacks seriously affect both the online world and the physical one. Although there are several methods of illegally accessing clients and servers, attackers gain accesses using malware in most of cases[1]. Web attacks are comprised of the following labels from CSE-CIC- IDS2018: "SQL Injection", "Brute Force-Web", and "Brute Force-XSS"[10].

2.1.4 Random Forest Algorithm

Researchers have suggested many Intrusion Detection strategies over the years to deal with the complexity and number of threats to computer systems that have grown over time. In the domain of behavior-based intrusion detection systems, Random Forest models have been delivering a noteworthy performance on their applications. Classification, feature choice, and proximity metrics are provided using particulars of the Random Forest model[8].

A collection of Decision Trees, the Random Forest model can be applied to classification or regression. In the classification situation, the forecast is based on the Decision Trees' majority vote for the projected values, while in the regression case; the prediction is based on the average of the trees' results[8]. During the training phase, a training set Ti is created for each tree, taking into account the samples from the initial training set T. To build each tree

split, m features are then chosen at random and evaluated by a measure to determine which one should produce the split. Different trees are provided by this randomness, and when combined, they typically produce higher prediction performance[8].

One of the most well-known tree-based machine learning techniques in hydrology is random forest (RF). A random forest is a collection of classification and regression trees that addresses the overfitting problems of individual decision trees while maintaining the predictive power of the trees. The method was created because of its adaptability and accessibility in widely used programs like R or MATLAB, it quickly gained popularity as a tool in many geoscientific fields[9].

2.1.5 Feature Selection

Along with the expansion of data sets, new data kinds have also emerged, including data streams on the Web, microarrays in genomics and proteomics, networks in social computing, and system biology. Researchers and practitioners are discovering that feature selection is a crucial element to successful data mining in order to use data mining technologies efficiently[5]. Feature selection is a crucial and popular dimensionality reduction strategy for data mining that involves choosing a subset of the original features based on specific criteria. It delivers immediate effects for applications, including as speeding up a data mining algorithm and enhancing mining performance like predicted accuracy and result comprehensibility, by reducing the number of features and removing irrelevant, redundant, or noisy data[5].

High-throughput computer-based techniques are rapidly advancing, giving people unrivaled potential to improve their production, service, communication, and research capacities. In the

meantime, massive amounts of high-dimensional data are accumulated, testing cutting-edge data mining techniques. Successful data mining applications require the use of feature selection, which can efficiently reduce data dimensionality by removing the unnecessary (and redundant) characteristics[6]. For decades, machine learning and data mining researchers have been working on the topic of feature selection. This research has been widely applied to many different domains, including genetic analysis, text mining, picture retrieval, and intrusion detection, to mention a few. Feature selection is crucial for real-world concept learning issues since it can hasten learning and enhance concept quality[6].

2.2 Theoretical Framework

In the context of data analysis and computing, artificial intelligence (AI), and in particular machine learning (ML), have expanded quickly in recent years. These technologies often enable applications to perform intelligently. The most widely used newest technologies in the fourth industrial revolution, machine learning (ML) often gives systems the ability to learn from experience and improve automatically without being specifically designed (4IR or Industry 4.0)[15]. The fields of machine learning and deep learning are subfields of artificial intelligence. The study of computer programs that use algorithms and statistical models to learn through inference and patterns without being explicitly programmed is known as machine learning. The last ten years have seen major advancements in the field of machine learning. In the last ten years, machine learning (ML) has emerged as one of the most revolutionary technological developments. The figure 2.1 below show the three types machine learning algorithm:

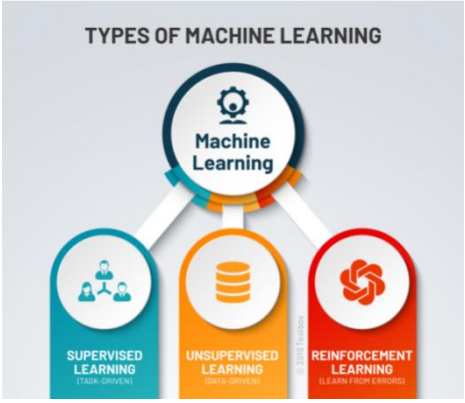


Figure 2.1: Machine Learning Algorithm Types

(Source: https://doi.org/10.1007/s42979-021-00592-x)

Supervised learning, unsupervised learning, and reinforced learning are the three broad categories into which machine learning algorithms can be roughly divided. The decision-making process can be automated using supervised learning algorithms, which are frequently regarded as the most effective ML algorithms. As opposed to supervised learning, unsupervised learning merely gives the algorithms input data without any known output data or goal variables[14]. The algorithms learn to recognize patterns from unlabeled, unknowable material through self-organization. As a result, unsupervised ML algorithms are harder to visualize and assess when trying to solve an issue. Unsupervised learning issues include mapping and aggregating wear zones based on the sliding speed and load during sliding interactions. Through trial and error, the model is trained to make a series of judgments in reinforcement learning. For instance, a self-driving automobile is trained via reinforcement learning. The ML model chooses among various choices to carry out a task while receiving feedback during the training phase. Through predetermined reward and punishment mechanisms, feedback is given[14].

Tackling classification and regression issues is the sole purpose of supervised learning in machine learning algorithms. There are numerous algorithms for solving classification problems, including NB (Naive Bayesian), SVM (Support Vector Machine), DT (Decision Tree), and others. The ensemble learning approach was created because it was clear that all of these single classifiers were prone to overfitting issues and performance bottlenecks. DT and Bayesian are more typical of single classifier technologies[8]. To some extent, these algorithms have aided in the advancement of categorization technology, and all facets of study and application have been thoroughly completed. However, a single classifier's performance increase has hit an intractable bottleneck due to its inherent restrictions, prompting people to suggest combining many classifiers. Combining multiple base classifiers yields a final classification result by integrating all classification outcomes. A multiclassifier combination called RF (Random Forest) is created using this background[9].

Use of the Random Forest algorithm, due to the RF algorithm's robust performance across the board, it is widely employed in various sectors. The RF algorithms to anticipate urban smog, apply it to environmental protection, and then examine and explain smog management techniques. The fund rating model was created using the RF algorithm, and they believed the information ratio to be the most significant indicator of fund evaluation, followed by a determinable coefficient. The study showed that the model's stability and accuracy had reached a very high standard[10]. The nonparametric RF method was first used and to predict the direction of fund excess returns in China. This work demonstrated that the RF method outperformed random walk and support vector machine algorithms and, to a certain extent, demonstrated the predictability of the domestic financial market. On-site monitoring and

penetration prediction of plasma arc welding using the RF algorithm and investigated the discriminant analysis approach and commuter identification based on RF of smart card data[10].

Numerous feature selection techniques have been proposed in the recent years. Filter, wrapper, and embedding approaches were the three categories into which Guyon and Elisseeff divided feature selection methods. Each attribute (feature) in the filter technique will have a calculated score, and all attributes with scores greater than a determinant cutoff value are selected, according to what they said. The chosen characteristics were the most instructive[7].

There are three categories of feature selection techniques: filter, wrapper, and embedding. Without taking into account the underlying learning scheme, filter approaches primarily concentrate on the characteristics of data examples. Wrapper techniques, embedded methods take feature selection into account during the training phase to decrease the time complexity of reclassifying various reductions[8].

Numerous articles have been written, all of which have compared various feature selection methodologies. Classification is one of the greatest ways to judge how well feature selection techniques operate. This assessment method has been used in numerous publications that have been published, and it works effectively[11]. Hall suggested a brand-new correlation feature selection (CFS) approach. The author demonstrated how to use this approach for classification, regression issues and talked about how crucial feature selection techniques are. They came to the conclusion that the traits they had chosen were autonomous and adequate for the learning process. For managing an evolutionary technique and developed a filter method. Their approach was contrasted with previous evolutionarily driven single-filter

approaches[11]. In terms of categorization accuracy, they tested their method and compared it to others and created a brand-new filter technique that combined the results of various filter techniques before contrasting them with those of single filter techniques. With regard to the quantity of attributes, they employed both small and large datasets[12]. The techniques were contrasted in terms of the classification accuracy metric and compared the classification accuracy of real datasets to compare wrapper and filter feature selection approaches on large datasets with respect to the validity of the selected features[13].

2.3 Review of Related Work on Web-Based Attack

Numerous researchers have looked into and improved Algorithms, Machine Learning Techniques, and Web-Based Attacks. Below, a few recent articles on this topic are discussed.

2.3.1 Defense Mechanism Using Multilayered Approach and SQL Injection Methods for Web- Based Attacks

The damaging effects of secure software development on various groups were highlighted in the article. Investigated about validation threats such buffer overflows, SQL injections, and cross-site scripting (XSS) and Came to the conclusion that no defensive strategy is completely foolproof. When unreliable content is sent through queries, SQL injection happens. Users' sessions are hijacked by XSS, which then sends users to fraudulent websites[16]. When a process or program tries to store more data than is intended, BOF (Buffer over Flow), it happens. The suggested model filters input for SQL injection and Cross-Site Scripting using detection modules for different attacks, validation, and analysis modules for control flow graph and validation flow graph, as well as semantic and syntactical validation, to counter attacker's malicious injection.

2.3.2 Detecting Web Attacks using Stacked Denoising Autoencoder and Ensemble Learning Methods

Web-based anomalies continue to pose a significant security risk to the Internet. In order to identify unusual HTTP requests, this study suggests combining the outputs from several Stacked Denoising Autoencoders (SDAEs) using Sum Rule and Xgboost[17]. Sum Rule and Xgboost inherit the distinctive advantage of SDAE, which does not necessitate the extraction of handcrafted features. Additionally, these techniques can deal with evolving web vulnerabilities, where malicious code is inserted into various request body and header fields. On the DVWA dataset and the dataset obtained from a real-world application, experiments were run[17]. In comparison to the most advanced Regularized Deep Autoencoders, Isolation Forest, C4.5 decision tree, and Long Short-Term Memory network, Sum Rule and Xgboost show to get higher F1-score.

2.3.3 Detecting Web Attacks with End-To-End Deep Learning

Three new insights into the study of autonomous intrusion detection systems are offered by the study. Consider first whether a method based on the Robust Software Modeling Tool (RSMT), which autonomously monitors and describes the runtime behavior of online applications, is feasible for detecting web attacks. The second section will go over how RSMT trains a stacked denoising autoencoder to encode and reconstruct the call graph for end-to-end deep learning, where a low-dimensional representation of the raw features with unlabeled request data is used to detect anomalies by computing the reconstruction error of the request data. Third, examine the outcomes of empirically testing RSMT on simulated datasets and real-world applications that have been intentionally vulnerable[18]. Results reveal

that even with little domain expertise and labeled training data, the suggested approach can quickly and accurately identify threats like SQL injection, cross-site scripting, and deserialization.

2.3.4 Detection of SQL Injection Attacks by Removing the Parameter Values of SQL Query

The detection of SQL injection attacks and safeguarding web applications against them are discussed in this work. Technology in the digital age anticipates attack-free solutions that increase data sharing security. The suggested solution makes systems safer by enabling web applications to recognize code injection (SQL injection) assaults before any data is lost. The best way to achieve the results anticipated from this suggested strategy is to combine existing SQL injection detection systems to construct more robust processes to increase the robustness of web applications[19].

2.3.5 Review of SQL injection attacks: Detection, to enhance the security of the website from client-side attacks

This paper main goal is to evaluate earlier research on SQL injection attacks and the dangers they pose to websites and applications. The other goal is to become familiar with the most recent research on SQL injection attacks' causes and remedies in order to prevent exposure to them and create a secure online environment. The SQL queries against the database are discussed to distinguish between malicious and legitimate ones. The most common technique used by hackers to obtain sensitive data and information from consumers is SQL injection[20].

The figure 2.2 below show SQL injection attack:

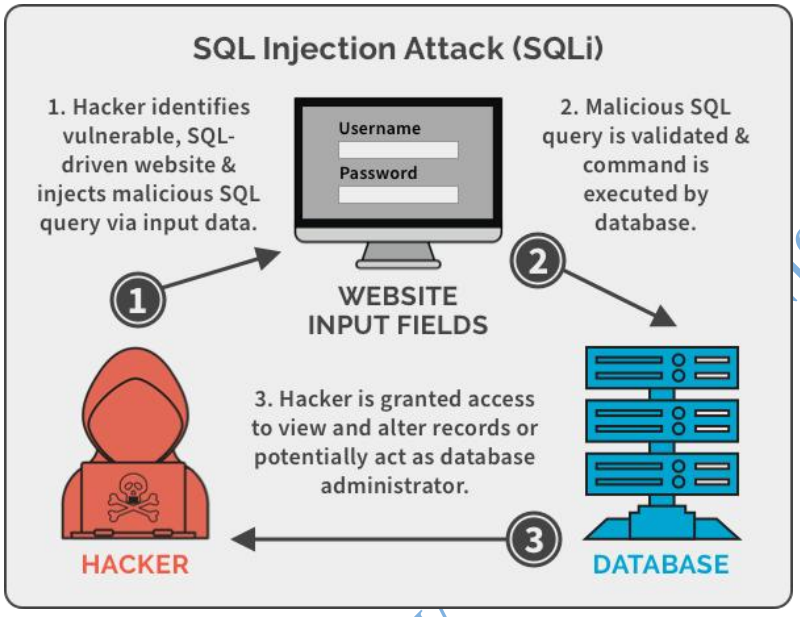


Figure 2.2: SQL injection attack Web-based application security

(Source: http://dx.doi.org/10.22075/ijnaa.2022.6152)

Examples of database risks and solutions include excessive privilege abuse, justifiable privilege abuse, privilege elevation, and platform vulnerabilities. Previous research has uncovered techniques for detecting malicious SQL injection attacks that are more accurate and take less time to catch. SQL queries are also examined[20].

2.3.5 Detection of Web Cross-Site Scripting (XSS) Attacks

By developing a program that recognizes XSS vulnerabilities by completely mapping internet applications, this work aims to enhance the functions currently offered by the internet for preventing XSS attacks. The uniqueness of this work lies in the inclusion of pre-approved XSS vulnerability scanning in examined internet functions along with the use of

environmentally friendly algorithms for locating exceptional XSS vulnerabilities in order to produce an exhaustive internet resource map. Internet utility protection is more effective when it uses the built program to find XSS vulnerabilities. Additionally, this program makes it easier to use web apps[21]. figure 2.3 show the percentage of websites vulnerable to different cyberattacks:

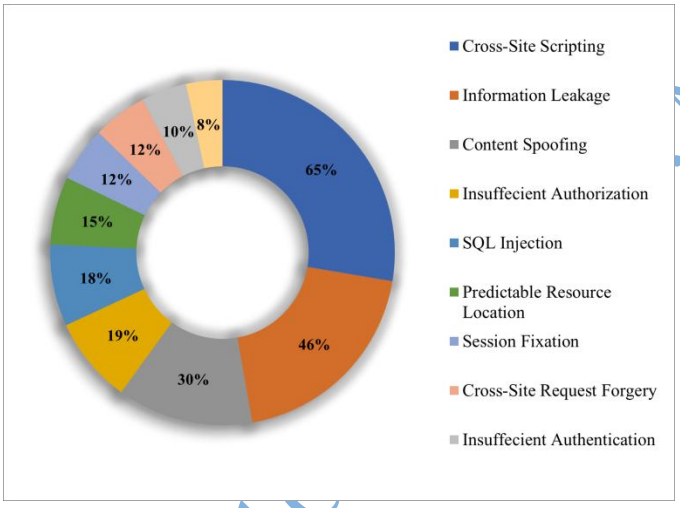


Figure 2.3: Percentage of Websites vulnerable to different Cyberattacks.

(Source: https:// doi.org/10.3390/electronics11142212)

By identifying XSS vulnerabilities by comprehensively mapping internet applications, the tool created for this research protects internet functions from XSS attacks. To find uncommon XSS vulnerabilities, this program uses the most environmentally friendly fixed software-based algorithmic methodology. For the purpose of creating an exhaustive map of helpful online resources, this application also analyzes previously authorized XSS vulnerabilities. The effectiveness of internet utility protection will rise with the discovery of XSS vulnerabilities. The created program makes cleaning applications easier as well[21]. This program's ability to generate documentation and suggest measures against the discovered

XSS vulnerabilities makes it usable even by users who are unaware with the fundamentals of information security.

2.3.7 Enhancement in Cloud Security for Web Application Attacks

The system in this study employs a novel strategy to guarantee authentication, confidentiality, and all content-based security features. Based on advertisements and subscriptions, identity encryption is utilized to assign credentials to publishers and subscribers. To remove third-party concerns about certificate status and lessen infrastructure needs, certificate-based encryption is used[22].

2.3.8 Ensemble Modelling for Predicting the Relation between Biopsychosocial Signals and Seizures using the Gradient Boosting Method

The methods for classifying EEG signals based on ensemble learning methods XGBoost and random forest and feature engineering using principal component analysis (PCA) in rapid miner for dimensionality reduction and Correlation-based Attribute Ranking for performance evaluation in the data set in bins are discussed in this paper. A performance-based analysis is completed and provided for evaluation in order to validate the findings. Based on evaluation performed on the same data set, results are compared with current models like Random Forest (RF) and presented with ROC curves and other characteristics for comparison[23]. This method shows that, when compared to other methods currently in use, ensemble learning techniques have a significant advantage in terms of practical value. These techniques can be used to advance the development of BCI technology, which could help people with mental health conditions like seizures, which can sometimes result in death if proper diagnosis and treatment are not provided.

2.3.9 Machine Learning based Intrusion Detection System for Web-Based Attacks

In order to discriminate between regular and abnormal traffic, we cleaned and labeled the CSIC HTTP 2010 dataset before conducting our experiments. In order to find missing features for a typical attack, the data was finely preprocessed using a Python script. Additionally, by using various Machine Learning classifiers like J48, Naive Bayes, OneR, and Decision tables that use evaluation metrics to find the accuracy using Weka 3.8, feature extraction from the dataset played a key role in identifying malicious behavior and the attack types like SQL injection (SQLi), Cross-Site Scripting (XSS), and Buffer Overflow[24]. Additionally, 20 features were extracted with enhanced web-based attack detection thanks to the use of fine-tuned feature set engineering, raising the true positive rate. Last but not least, the J48 decision tree algorithm was shown to be the top performing algorithm with the best attack detection rate of 94.5% in our testing results using three machine learning algorithms (J48, Naive Bayes, and OneR).

2.3.10 SQL – Attacks, Modes, Prevention

By reviewing a number of the resources and completed research, the study aims to put SQL attacks into perspective. With the input test queries, the suggested compound approach correctly identifies the SQL injection attack. Test queries like TEST' or 1=1-- are used to determine whether or not a website has SQL injection[25]. The attacker is prohibited from accessing the database using such specifically created cunning queries that could lead to an unauthorized access and allow changes to be made to sensitive data in the database. The usage of parameterized queries assists in preventing such problems. It has been determined that the suggested approach works well in preventing SQL injection attacks.

2.3.11 Applications of Machine Learning Predictive Models in the Chronic Disease Diagnosis

In this work, the most recent techniques for using ML models in the first CD diagnosis were examined. The search for the documents included in this study came from the libraries of PubMed (Medline), Cumulative Index to Nursing and Allied Health Literature (CINAHL), and 453 publications that were published between 2015 and 2019. In the end, 22 papers were chosen to accurately show all modeling techniques, including CD diagnosis and usage models of distinct illnesses with related strengths and limitations[26]. The findings imply that, while each method has advantages and disadvantages of its own, there are no established techniques for identifying the optimum strategy in actual clinical practice. Support vector machines (SVM), logistic regression (LR), and clustering were the most frequently employed techniques. These models are predicted to play a larger role in medical practice in the near future because of their great applicability in the categorization and diagnosis of CD.

2.3.12 An Anomaly Detection Method to Detect Web Attacks Using Stacked Auto-Encoder

In this study, a feature learning method based on deep neural networks and an isolation forest classifier are developed. On the CSIC 2010 data set, this strategy was contrasted with others that do not use feature extraction models. Employed several activation functions and learning techniques to our deep neural network. Results indicate that deep models are more accurate than feature-free techniques[27]. Used the SAE approach to extract pertinent information from HTTP request logs for anomaly detection in firewalls for online applications and to

differentiate between abnormal and normal data, Isolation Forest has been employed as a one-class learner.

2.3.13 Approach to Intelligent Monitoring of Cyber Attacks

The construction of intelligent automated systems for monitoring items in the web space can be based on the concepts and methodologies based on neural fuzzy solutions that are proposed in this study. It is suggested that decision-making processes be based on the formalization of prior expert knowledge of fuzzy based production norms[28]. The potential of a neuro fuzzy classifier in the form of a three-layer fuzzy neural network is examined in the context of solving the issues of classification and extension of classification of input data on the features of the dynamics of attributes of the object of monitoring.

2.3.14 On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks

Learn more about the effects of performing SIDS using predetermined rulesets in this paper. Investigate experimentally how three SIDS behave when facing web attacks. Use seven attack datasets to evaluate, in particular, the detection rate attained using specified subsets of rules for Snort, ModSecurity, and Nemesida. Using a significant trace from a public website, the study additionally ascertains the accuracy and rate of warning generated by each detector in a real-life situation[29]. The results demonstrate that the maximum detection rate attained by the SIDS under test is both below what is anticipated for known threats and insufficient to properly safeguard systems. The findings also show that each detector's detection capabilities and false alarm rate are significantly influenced by the predetermined settings that have been engaged on each one. With the less-sensitive preconfigured ruleset activated, Snort and

ModSecurity either achieved a very low detection rate or a very low precision (activating the full ruleset). Additionally, it was discovered that utilizing different SIDS for a group decision could increase either the precision or the detection rate, but not both.

2.3.15 Cross Channel Scripting and Code Injection Attacks on Web and Cloud-Based Applications: A Comprehensive Review

Cross channel scripting threats and assaults, which are among the most dangerous online application vulnerabilities, were covered in this review study. This has been found to be a serious issue for the web apps of today. Examined eight distinct categories of consumer networking devices from a number of vendors and discovered that each one has significant XCS issues. XCS is a result of obsolete software libraries and embedded devices with smart capabilities[30]. Furthermore, these devices are typically vulnerable to external attacks because of the numerous Internet protocols. This article also identified research gaps and described various state-of-the-art cross channel scripting attack-based mechanisms. In order to detect and counter cross channel scripting attacks and their variants, current online applications use a variety of strategies, techniques, and tools, all of which are listed in this study article. It is determined that each embedded device is audited at three stages. Initially, a general analysis was completed using NMap, an open-source program with a free auditing and network detection utility.

2.3.16 A Survey on SQL Injection Attacks: Detection and Prevention

The threat posed by SQL injection attacks to programmers and web applications is steadily growing. These attacks can be launched in a variety of ways, but they can also be detected and avoided in a variety of ways. In past studies, the researchers discovered detection

systems, but they are no longer useful due to the severity of the attacks. Here, they provided a thorough assessment of the literature on various SQL injection strategies[31]. Additionally, they covered a variety of techniques for detecting and preventing this attack, including input validation, AES, DES, machine learning, etc. This study assists academics and programmers who wanted to learn about all the problems that still plague web applications and which strategies can be used to identify and prevent SQL injection attacks. It can also help the general public understand SQL and its hazards.

2.3.17 Your WAP Is at Risk: A Vulnerability Analysis on Wireless Access Point Web-Based Management Interface

The following major query is addressed in this work: Are there any faults or vulnerabilities in the modern, off-the-shelf wireless access points' Web-based management interfaces. It reveals a large number of various medium-to-high severity flaws that can be easily or indirectly exploited. In total, 28 zero-day attacks across 13 types of vulnerabilities are revealed[32]. Findings include HTTP request smuggling and splitting, replay, denial of service, and information leakage, among others. They also include legacy path traversal, cross-site scripting, and clickjacking threats. The attacker can gain access to the administrator's (admin) login information and the WAP's Wi-Fi passphrases, or lock the admin out of the Web interface permanently. In addition to everything else, the report describes the further countermeasures needed to address the holes found as well as the hardening measures already implemented by these devices.

2.3.18 Information Security threats and attacks with conceivable counteraction

This study provides a thorough analysis of various information security threats and assaults, which are divided into three categories: network, host, and application. The study then discusses attacks and security requirements with several potential mitigation approaches, taking hacking phases and methods as a framework, and further illustrates the relationship between attacks in systems and assaults in information[33].

2.3.19 Survey of Attack Projection, Prediction, and Forecasting in Cyber Security

This study offers a survey of forecasting and prediction techniques applied to cyber security. This survey compares and contrasts strategies based on continuous models like time series and grey models with those based on discrete models like attack graphs, Bayesian networks, and Markov models. Discuss machine learning and data mining techniques in more detail, which have received a lot of attention recently and seem promise for the dynamic context of cyber security. The poll also emphasizes issues with the approaches' practical applicability and issues with their evaluation[34]. The issue was framed within the framework of studies on cyber situational awareness and intrusion detection. Taxonomy of techniques was offered, and each subcategory was thoroughly examined. The final assessment compared the approaches and included associated issues and acquired lessons.

2.3.20 Prevention and Fighting against Web Attacks through Anomaly Detection Technology. A Systematic Review

In particular, using the standard method of a systematic review of literature in the field of computer science, proposed by Kitchenham, to conduct a systematic review of the application of anomaly detection techniques in the prevention and detection of web attacks.

This approach used on a group of 88 publications that were selected from 8041 reviewed papers in total and published in reputable journals. In order to determine the current state of the art in web anomaly detection, this paper covers the methodology used in this systematic review as well as the outcomes and conclusions attained[35].

2.3.21 Robust Early Stage Botnet Detection using Machine Learning

The identification of features for the detection of a botnet in dispersed cyberattacks was the main focus of this study. The C&C channel, which is an early stage of the botnet life cycle, was used in the study to test the early-stage detection methodology. The proposed method, which is known as centralized architecture, concentrated on IRC and HTTP protocols during the C&C channel. The suggested method chooses the best feature to find the botnet attack. Experimental findings demonstrate that the suggested technique yields superior outcomes. In comparison to the current approach, the proposed approach had accuracy of 99%, TPR of 0.99%, and FPR of 0.007% using 40 features and accuracy of 97.8%, TPR of 0.97%, and FPR of 0.02% using 37 features[36].

2.3.22 An efficient metaheuristic algorithm-based feature selection and recurrent neural network for DoS attack detection in cloud computing environment

In order to address these kinds of problems, this paper suggests an effective DoS attack detection system that makes use of the Oppositional Crow Search Algorithm (OCSA), which combines the Crow Search Algorithm (CSA) and Opposition Based Learning (OBL) technique. The proposed method consists of two stages: feature selection using OCSA and classification using a classifier based on recurrent neural networks (RNNs). The OCSA algorithm is used to choose the key features, which are subsequently provided to the RNN

classifier. In the subsequent testing procedure, the RNN classifier is used to categorize the incoming data[37]. It makes sure that compromised data is removed and that standard data is separated (and preserved in the cloud). The experimental assessment findings using the benchmark data set show that the suggested strategy beats the other traditional methods in terms of Precision, Recall, F-Measure, and Accuracy by 98.18%, 95.13%, 93.56%, and 94.12%, respectively.

2.3.23 IoT malicious traffic identification using wrapper-based feature selection mechanisms

In this study, effective features were initially selected using a bijective soft set, and then a novel CorrACC feature selection measure was developed. Then, using a wrapper technique to filter the features and choose the most useful features for a certain ML classifier by utilizing the ACC metric, a new feature selection algorithm termed CorrACC was conceived and built. On the BoT-IoT dataset, four different ML classifiers were employed to evaluate the proposed techniques. Our algorithms' experimental results have shown promise and have a 95% accuracy rate[38]. It is evident that the suggested methods were successful in identifying the best feature set, anomalies, and intrusion points in the IoT network. It is also obvious that ML classifiers can effectively classify both IoT attacks and regular traffic without changing the training dataset. Additionally, the proposed approach achieves excellent results in terms of accuracy, precision, sensitivity, and specificity metrics using the features that it has chosen.

2.3.24 Nscanner: Vulnerabilities Detection Tool for Web Application

The study suggested a program called Nscanner, which aids most web developers and security auditors (pentesters) in evaluating their web applications. Due to its automated capabilities, it can quickly analyze all websites for serious vulnerabilities like SQLi and XSS. When the detection is successful, a report will be produced so the user can review the findings. A regular user can check any harmful file's content to find malware programs in addition to performing web assessments. A report containing the findings of the malware detection will be produced for user analysis. In the end, this research might also contribute to increasing cyber security awareness among Malaysia's vast majority of internet users[39]. Most working individuals and non-working adults must increase their knowledge and expertise in self-defense against cyberattacks. Technology should always be a major factor in thwarting the majority of malevolent online criminal behavior.

2.3.25 Building an efficient intrusion detection system based on feature selection and ensemble classifier

In this research, the study proposes a new framework for intrusion detection that is based on feature selection and ensemble learning methods. The CFS-BA heuristic algorithm, which chooses the best subset based on the correlation between features, is presented as the initial stage for dimensionality reduction. Afterwards, provide an ensemble strategy that incorporates the C4.5, Random Forest (RF), and Forest by Penalizing Attributes (Forest PA) algorithms. Finally, the probability distributions of the base learners are combined for attack recognition using the voting mechanism[40]. The suggested CFS-BA-Ensemble method is able to demonstrate greater performance than other related and state of the art approaches under a number of criteria, according to the experimental results using the NSL-KDD, AWID, and CIC-IDS2017 datasets. The obtained results for the AWID dataset show accuracy of 99.52%

and 0.15% FAR with a subset composed of just 8 features, while the experimental results for the NSL-KDD dataset show accuracy of classification equal to 99.81%, 99.8% DR, and 0.08% FAR with a subset of 10 features. Surprisingly, our model, when applied to the subset of 13 characteristics from the CIC-IDS2017 dataset, achieves the greatest accuracy of 99.89% and DR of 99.9%.

2.3.26 Improving Detection Accuracy for Imbalanced Network Intrusion Classification using Cluster-based Under-sampling with Random Forests

Through the use of cluster-based under-sampling and a Random Forest classifier, this research proposed a new technique for increasing detection rate to categorize minority-class network attacks/intrusions. A multi-layer classification strategy is the suggested solution, which can process the extremely unbalanced huge data to accurately identify known or unidentified network breaches. The proposed technique initially determines whether a data point or incoming data is an attack or intrusion (as opposed to normal behavior), and if it is, the proposed method is used[41]. This paper's main goal is to improve the classification of imbalanced network intrusions for low-frequency attack detection accuracy. Evaluated the effectiveness of the suggested strategy in comparison to common data mining algorithms. The hybrid technique that has been proposed increases detection rates while decreasing false positive rates, according to experimental findings on the KDD99 benchmark dataset.

2.3.27 Clustering Enabled Classification using Ensemble Feature Selection for Intrusion Detection

In this study, an ensemble feature selection method and an anomaly detection method are proposed. These methods integrate supervised and unsupervised machine learning techniques to classify network data and find previously unidentified attack patterns. In order to achieve this, an ensemble model that selects 8 common features makes use of three alternative feature selection methodologies. Additionally, the training cases are divided into k clusters using the Manhattan distance before employing k-Means clustering[42]. The generated clusters, which reflect a density zone of normal or anomalous examples, are then used to construct a classification model. This in turn aids in assessing how well the clustering works to identify potential attack patterns in the data. Our classifier's performance is assessed using data from the Kyoto dataset, which was gathered between 2006 and 2015. By doing various tests with the Kyoto 2006+ dataset, which was created during a 9-year period of real traffic data collecting (between November 2006 and December 2015) from various types of honeypots at Kyoto University, the performance was assessed and compared.

2.3.28 Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning

The Smart Detection system, an online method of DoS/DDoS attack detection, has been introduced in the article. Based on samples obtained by the sFlow protocol directly from network devices, the software employs the Random Forest Tree method to categorize network traffic. To calibrate and assess system performance, a number of experiments were conducted. The proposed method has enhanced performance when compared to certain recent and pertinent methodologies that are available in the literature, according to the results. The suggested system was tested using the CIC-DoS, CICIDS2017, and CSE-CIC-IDS2018

intrusion detection benchmark datasets. It was found to be capable of classifying several DoS/DDoS attacks, including TCP flood, UDP flood, HTTP flood, and HTTP slow[43].

2.3.29 Internet of Things Cyber Attacks Detection using Machine Learning

The goal of this study was to examine and identify IoT network attacks using machine learning techniques. The Bot IoT was utilized as a dataset in this study because of its frequent updates, substantial attack diversity, and variety of network protocols. From the raw traffic traces, flow-based features were extracted using the CICFlowMeter. The dataset's 84 network traffic features generated by CICFlowMeter define the network flow. The Random Forest Regressor algorithm was utilized during implementation to determine the importance of weight calculations and which features would be used in the machine learning techniques. These computations were performed using two different strategies[44]. In the first method, the importance weights for each type of attack were calculated separately. In the second method, all attacks were gathered into one group and the importance weights for this group were calculated, i.e., the common characteristics that were crucial for all attacks were identified. Seven machine learning algorithms that are popular and have a variety of attributes were then applied to the data[44]. Following are some algorithms and the F-measure-achieved performance ratios: F-measure had a value between 0 and 1, Naive Bayes, QDA, Random Forest, ID3, AdaBoost, MLP, and K Nearest Neighbors all had values between 0.77 and 0.97.

2.3.30 Hybrid feature selection technique for intrusion detection system

In this work, a hybrid feature selection approach that may effectively integrate the advantages of the filter and wrapper selection processes is used. It is anticipated that the

possible hybrid solution will choose the best combination of attributes for detecting intrusion. Correlation feature selection (CFS), together with the best-first, greedy stepwise, and genetic algorithms, were used to search for the proposed hybrid model. Each feature that was initially chosen using the filter approach is evaluated by a random forest (RF) classifier as part of the wrapper-based subset evaluation. The reduced feature selection on the KDD99 and DARPA 1999 datasets was evaluated using the RF algorithm in a supervised setting with ten-fold cross-validation. The experimental results demonstrate that the results of the hybrid feature selections were satisfactory[45].

2.3.31 Feature-Selection-Based Ransomware Detection with Machine Learning of Data Analysis

In this study, a system comprised of an Argus server and client applications was designed, and a flow-oriented method known as Biflow was suggested in order to create a ransomware detection module based on open malicious network traffic datasets. The data size was being decreased by these datasets to 1000:1. The goal of this study is to determine whether the feature selection had an impact on the classification precision rate by combining six feature selection algorithms to examine the preselected column and its classification-related relevance and noticed a gradual reduction in the input features. Along with an increase in accuracy, the decision tree's node count also shows a decline. Last but not least, fewer rules improve intrusion detection performance, cut down on processing time, and lead to an earlier detection of abnormal traffic[46].

2.3.32 Intrusion Detection System with Recursive Feature Elimination by using Random Forest and Deep Learning Classifier

According to the study, the attacker is not need to know the IP address or the Flow ID. Similar to this, timing and Destination IP information are not employed in the model's training stage because it is impossible to foresee when and how assaults might be carried out[47]. The Python Keras framework is used in the first phase to create a deep learning - DMLP model that can exploit the dataset's features. The input layer of the produced model is configured to have 80 characteristics. In order to create a single output value in the output layer, the input layer is intended to be incrementally lowered in the hidden layers.

2.3.33 Preventive Measures for Cross Site Request Forgery Attacks on Web-based Applications

This study examined various techniques that may be used to check for the CSRF vulnerability and shed light on them. The research's primary goal is to find the available defenses against CSRF assaults. The best tool can be found by examining the methodologies used in each of the solutions. After incorporating the fixes into the web application, tests were run to see how well each fix worked against the exploitation of the vulnerabilities. The study also suggests a combination method that combines passing a token through a hidden field and server-side validation with passing a token through the URL[48].

2.3.34 Web Application Attacks Detection Using Machine Learning Techniques

The study demonstrated and investigated machine learning strategies that can enhance MODSECURITY's detection capacities in terms of a decrease in false positives and an increase in true positives. Additionally, the study discussed the issue by pointing out various scenarios based on the accessibility of training data. The scenarios range from the uncommon,

but ideal, situation in which we have a dataset with actual application traffic to more realistic situations in which we have only valid requests to an application that could be gathered, for example, during the functional testing phase[49]. The absence of publicly accessible labeled datasets with complete HTTP requests was one of the biggest problems encountered. Only three datasets could be found, and only two of them were used in our experiments (see section III-B). The third dataset, the 1998 DARPA Intrusion Detection Evaluation Data Set, was discarded because it was built using network traffic rather than just web application traffic. These datasets are also at least ten years old. And believe that those datasets no longer accurately reflect the state of the relevant technologies today.

2.3.35 An Intrusion Detection System Using Machine Learning Algorithm

In order to assess the efficacy of these algorithms by categorizing these attacks into their various classes, the KDDCup99 Test datasets were analyzed using the machine learning algorithms Bayes Net, J48, Random Forest, and Random Tree. This research is conducted using a positive research methodology. The experimental findings demonstrate that the Random Forest and Random Tree algorithms seem to be the most effective when applying the classification technique to the Test dataset. Weka is the experimental tool used, and the parameters used for the computation are Precision, Recall, and F-measure. WEKA is used to perform a correlation-based feature selection on the dataset with a Best First search method[50].

2.3.36 Anomaly-Based Network Intrusion Detection System through Feature Selection and Hybrid Machine Learning Technique

The study suggested a feature selection, K-Means clustering, and XGBoost classification model-based anomaly-based network intrusion detection system, and tested the performance

of proposed system using the NSL-KDD dataset and the KDDTest+ dataset. A reduced feature subset of the NSL-KDD dataset is created using a feature selection method based on attribute ratio (AR). Hyperparameter adjustment of each classification model corresponding to each cluster is implemented once K Means clustering has been applied. For the KDDTest+ dataset, the suggested model achieves accuracy of 84.41%, detection rate of 86.36%, and false alarm rate of 18.20% using just 2 clusters[51]. The suggested model outperforms deep neural networks (DNNs) based on recurrent neural networks (RNNs) and other tree-based classifiers in terms of performance. Additionally, due to feature selection, the suggested model uses only 75 out of 122 features (61.47%), which is less than the entire number of features needed to train the model to reach this level of performance.

2.3.37 An Intrusion Detection System for Web-Based Attacks Using IBM Watson

In order to strengthen the security of web systems, machine learning algorithms are utilized in this study to train models to categorize these requests. The CSIC 2010 dataset provided the data for the training and testing in this study[52]. Testing was done on the J48, Naive Bayes, OneR, Random Forest, and IBM Watson LGBM algorithms. T-rate, precision, recall, and f measure were the measures employed. The outcomes demonstrated that, when compared to other algorithms in the literature, the algorithm utilized by the Watson tool (LGBM) performed the best across all measures.

2.3.38 A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks

In order to increase the effectiveness of SQL injection protecting measures by lowering the false-positive rates in SQLi detection, this research explores the potential of applying data mining methodologies. The suggested method automates the classification of SQLi by first extracting features with CountVectorizer and then applying several supervised machine-

learning models[53]. among the available models, the one that provides the most accurate results has been picked. In order to lower the false-positive and false-negative rate, a new model called PALOSDM (Performance analysis and Iterative Optimisation of the SQLI Detection Model) has also been developed. The accuracy of the detection rate has also increased dramatically from a baseline of 94% to 99%.

2.3.39 Cross-Site Scripting Attacks and Defensive Techniques: A Comprehensive Survey

With a specific focus on the researchers' defensive techniques for preventing XSS attacks, this study has concentrated on revealing and comprehensively analyzing XSS injection attacks, detection, and prevention. It has divided these techniques into five categories: machine learning techniques, server-side techniques, client-side techniques, proxy-based techniques, and combined approaches[54]. The bulk of current cutting-edge XSS defensive strategies that have been thoroughly examined in this study provide defense against the classic XSS attacks, like stored and reflected XSS. The recently identified DOM-based and mutation-based XSS threats are not yet adequately protected against by any trustworthy solution. A combination of static, dynamic, and code auditing together with secure coding and ongoing user awareness campaigns concerning XSS arising at tacks are advised after reading all of the proposed models and pointing out their shortcomings.

2.3.40 Detect Cross-Site Scripting Attacks Using Average Word Embedding and Support Vector Machine

In order to detect web-based XSS assaults, this study provides an NLP-SVM model employing the average word embedding method. The approach makes use of NLP for the

detection task and the SVM model for processing text payloads at tacks. The detection model has been demonstrated to be effective in achieving improved accuracy, a spectacular detection rate, and low False Positive and Negative rates. A sizable dataset was used for both training and testing the NLP-SVM model[55]. The proposed model has been put through a number of analyses to be tested at different phases. In comparison to eight ML algorithms, the experimental findings strongly support the efficacy and idealism of the NLP-SVM technique, demonstrating substantial perfection and harmony of performance on numerous measurements of both classes. Additionally, the suggested model outperforms all other models in every way.

2.3.41 DeepWAF: Detecting Web Attacks Based on CNN and LSTM Models

This study offered a DeepWAF prototype implementation to identify online attacks using deep learning techniques. It also discussed the best way to use the widely used CNN and LSTM models, as well as their CNN-LSTM and LSTM-CNN combinational models. The experimental findings on the HTTP DATASET CSIC 2010 dataset show that all four types of proposed detection models produce acceptable results, with a detection rate of roughly 95% and a false alarm rate of roughly 2%[56]. Additionally, case studies were conducted to examine the reasons behind false positives and false negatives, which can be used to make future improvements. Work further demonstrates that the subject of web assault detection offers promising application opportunities for machine learning.

2.3.42 OwlEye: An Advanced Detection System of Web Attacks Based on HMM

OwlEye, a hybrid attack detection sensor based on Hidden Markov Model, was proposed in this study (HMM) [57]. It is made to defend against cross-scripting and SQL-injection web-layer code injection attacks. By utilizing both legitimate and malicious traffic in model

training, its novel bidirectory scoring architecture design has the advantage of achieving a satisfactory detection rate at a manageable false positive rate.

2.3.43 A Robust System for Detecting and Preventing Payloads Attacks on Web-Applications Using Recurrent Neural Network (RNN)

An RNN model was trained for this study's research using a dataset that included examples of various types of attacks on online apps. The XSS, SQLi, and Shell attacks are among them. The problem of a highly unbalanced dataset was solved using a random oversampling approach, which prepared the dataset for preprocessing. After resolving the imbalance issue, the dataset underwent pre-processing by going through data cleaning and tokenization. The tokenized data was transformed into an array which was used in feeding our RNN model as input[58]. The proposed model was trained over a period of two (2) epochs, with each epoch displaying the accuracy and loss values that the model was able to achieve for both the training and test sets of data. After training, the suggested RNN model provided us with testing data accuracy of 99.96% and training data accuracy of 99.91%. Additionally, a python flask was used to deploy our RNN model to the web in order to create a reliable system for identifying and preventing various payload assaults on online apps. The scope of this study is web application attacks only.

2.3.44 A Study on XSS Attacks: Intelligent Detection Methods

This study looked at and evaluated several performance metrics for XSS attack detection systems. This research reviews a number of papers published in well-read journals between 2019 and 2020 that meet these standards[59]. The reviewed articles are contrasted in terms of the performance metrics, nature, and simplicity of the algorithms. The research made the

assumption that using simple strategies to identify XSS attacks was preferable to using suggestions for specific artificial intelligence technologies.

2.3.45 Robust Training for Injection Attacks Detection in Web-based Applications

This study put forth an algorithm that incorporates a matrix learning loss term into the objective function of the initial neural network. The new loss acts as a regularization term that projects traffic data into a Euclidean space where distance can be used directly to measure the similarity of legitimate traffic and malicious traffic and push similar traffics toward each other and dissimilar traffics away from their false classes[60]. A projected gradient descent noise was also suggested to be added during the training phase to create a classifier that was broader. The experimental findings on the CIC-IDS2018 dataset demonstrate that introducing our projected gradient descent noise into the machine learning training phase and training neural network classifier with matrix learning loss term proves to be more robust in detecting tiny mutants of existing injection at tacks.

2.3.46 Web Abuse Using Cross Site Scripting (XSS) Attacks

Draw a generalized picture of XSS attacks and the various varieties in this study. Talk about other alternative code avoidance methods, such as strong defenses. The study includes explains discussion of relevant research that researchers have concluded regarding mitigating scenarios and strategies that might be used for prevention[61].

2.3.47 A Quick Review of Machine Learning Algorithms

The goal of this study was to review the most popular machine learning algorithms used to address classification, regression, and clustering issues. These algorithms' benefits and drawbacks have been examined, and different algorithms have been compared (where possible) in terms of performance, learning rate, etc. Additionally, it has been suggested how these algorithms can be used in real-world situations. We've talked about three different categories of machine learning techniques: semi-supervised learning, unsupervised learning, and supervised learning[62]. It is anticipated that it will provide readers with the knowledge they need to make an informed choice when determining the possibilities for machine learning algorithms that are accessible and then choosing the best machine learning algorithm for the particular problem-solving situation.

2.3.48 A Systematic Literature Review on Supervised Machine Learning Algorithms

In this study, 305 publications from three digital libraries/databases were initially collected from this systematic literature review investigation. 137 publications were excluded based on the exclusion and inclusion criteria in the first phase, and 76 studies were also excluded after a second evaluation of full papers, leaving only 61 papers to be included in this study[63]. It's fascinating to see that SML has primarily been used to categorize spam and text as well as in classification research for the healthcare and medical industries. The top two performance metrics for the SML algorithms in this study are SVM and ANN.

2.3.49 Machine Learning Algorithms - A Review

This research provided a brief overview and outlook on the numerous uses of machine learning techniques. The study of algorithms and statistical models that computer systems employ to carry out a particular task without being explicitly taught is known as machine

learning (ML) [64]. There are several daily-used programs that incorporate learning algorithms. One of the reasons an online search engine like Google works so well every time it is used to search the internet is because of a learning algorithm that has mastered the art of ranking web sites. These algorithms are used for a number of different tasks, including data mining, image processing, predictive analytics, etc. The main benefit of machine learning is that once an algorithm understands how to use data, it can carry out its work autonomously.

2.3.50 A Comparative Performance Assessment of Ensemble Learning for Credit Scoring

The main finding of this study is an experimental addition to the discussion about the best models for predicting credit risk. Five ensemble algorithms—RF, AdaBoost, XGBoost, LightGBM, and Stacking—as well as five conventional individual learners—NN, LR, DT, SVM, and NB—are evaluated side by side. The real-world credit dataset used for this research comes from Lending Club in the United States. Except for AdaBoost, experimental results show that ensemble learning produces clearly higher performance than individual learners. This contrasts with the previous hypothesis, and is likely due to the model's overemphasis on examples that are noise as a result of the training data's overfitting[65]. Five performance criteria—ACC, AUC, KS, BS, and model operating time—are best met by RF, with XGBoost and LightGBM coming in second and third, respectively. With regard to the majority of evaluation measures, LR beats the other classifiers among the five base learners. Additionally, the time cost of the task has been taken into account; NN and SVM require a lot of time, and the operating time of Stacking depends on the choice of the base models. Overall, under the limitations of specific time and hardware, RF, XGBoost, LightGBM, and LR may be the first and best option for financial institutions when it comes to credit scoring[65].

2.3.51 Performance Evaluation of Feature Selection and Tree-Based Algorithms For Traffic Classification

In this study, data analysis and exploration approaches were utilized to pick the most pertinent attributes that might be applied to categorize network traffic. The following step involved doing an empirical investigation of several DT-based classical classifiers (DT, RF, AdaBoost) as well as the more contemporary CatBoost, Light-GBM, and XGBoost classifiers. The data subset used for this comparison was chosen using Recursive Features Elimination (RFE). We have developed a method to determine the top 15 features out of 87 features in our dataset using RFE. This has not only drastically decreased the execution time but has also helped to improve the accuracy of the ML models by identifying useful features for network traffic classification in a real-world dataset[66]. Additionally, analytical and experimental results have demonstrated that adding additional characteristics does not always enhance classification performance. Furthermore, from a comparison of DT-based models, we draw the conclusion that, unlike DT and particularly RF, which generalize well with the default hyper-parameters, the hyper-parameter search is required to build accurate boosting-based models.

2.3.52 Feature Selection to Increase the Random Forest Method Performance on High Dimensional Data

This study demonstrated that the Random Forest technique's classification process may be sped up and its accuracy increased by using the BestFirst method to select the Correlation-based feature selector (CfsSubsetEval) feature. This has been demonstrated by a previous test on the high-dimensional Parkinson dataset, high-dimensional CNAE-9 dataset, and high-

dimensional Urban Land Cover dataset[67]. Execution times on average rise from 0.27 to 2.81 seconds. When examined on the Parkinson and Urban Land Cover dataset, the Random Forest method's average accuracy with feature selection grows along with its average speed. However, the average accuracy decreased when tested with CNAE-9 data.

2.3.53 A Comparative Analysis of Classification Algorithms on Diverse Datasets

In this study, three different classification methods were applied to ten datasets. The datasets have been chosen based on the size, quantity, and kind of their attributes. Results have been reviewed using various performance evaluation metrics, including ROC Area, mean absolute error, relative absolute error, F-measure, and Kappa statistics[68]. The accuracy, precision, and F- measure performance evaluation metrics have been used to do comparative study. For the various nature datasets and outline the characteristics and constraints of the categorization algorithms.

2.3.54 A Comparison of Feature Selection Algorithms for Human Activity Recognition

The UCI dataset is utilized in this investigation. Numerous Feature Selection (FS) algorithms are used in this sector. While reducing model complexity and dataset dimensionality, FS improves model performance. As a result, using properly selected FS algorithms, the best characteristics are found. Recursive Feature Elimination (RFE), Least Absolute Shrinkage and Selection Operator (LASSO), and Random Forest (RF) were the three FS algorithms that were put to the test[69]. These FS methods are compared using three particular classifiers. Metrics for performance evaluation thoroughly assess each algorithm's performance and identify the best one. The tools PYTHON and RSTUDIO are used to present the investigation's findings.

2.3.54 Comparative Study of Supervised Algorithms for Prediction of Students Performance

For three datasets, several classifiers including C5.0, J48, CART, Naive Bayes, SVM, KNN, and Random forest are developed to address issues in student performance prediction. Datasets for this study were gathered from e-learning platforms, colleges, and schools. Each dataset is subjected to Pearson correlation, and features that have a strong association to the desired result are picked[70]. The intellectual and behavioral elements that were identified as having substantially associated characteristics. G1 and G2 for dataset 1.Internal evaluation and attendance have an impact on student performance in dataset 2's 10th and 12th percentiles. The number of hands raised and the number of resources visited exhibit a good association in dataset 3. The different classifiers are applied to selected parameters along with their tuning parameters, such as the number of trials and confidence factor in C5.0, the

minimum number of instances per leaf in J48, the feature selection method in CART, the Laplace transform for Naive Bayes, the value of k in KNN, the number of trees to be formed in Random forest, and the type of kernel in SVM[70].

2.3.54 Performance Evaluation of Multiple Classifiers for Predicting Fake News

The accuracy of 14 distinct classifiers' performance was evaluated in this study, and we also tried to identify the stacks of classifiers whose performance stands out among the others based on our data. Extra Trees, Gradient Boosting, Logistic Regression, Passive Aggressive, SGD, Perceptron, Ridge, Linear SVC, Random Forest, AdaBoost, Decision Tree, SVC, Bagged, and KNeighbors Classifiers are some examples of classifiers[71]. It was found that the accuracy levels of Passive Aggressive, SGD, Perceptron, Ridge, and LinearSVC are higher than those of the others, surpassing 93%. The Rest 9 classifiers' accuracy is not particularly noteworthy.

2.3.55 Evaluation of Correlation Feature and Random Forest for Network Intrusion Detection

It is essential to have a network intrusion detection system. An appropriate classifier is necessary for a successful system. For greater accuracy, the classifier must only use a predetermined set of features[72]. Here, CFS and Random Forest feature selection techniques are discussed. It is clear that each strategy has advantages and drawbacks of its own. The greatest accuracy of CFS is 95.43%, while the maximum accuracy of Random Forest is 98.84%.

2.3.56 A Comparative Performance Evaluation of Random Forest Feature Selection on Classification of Hepatocellular Carcinoma Gene Expression Data

Hepatocellular carcinoma, one of the malignancies that result in death worldwide, was the subject of this study. Microarray data gene expression from 40 samples of hepatocellular carcinoma was collected from the National Center for Biotechnology Information website. This study's major goal is to assess how well various classification algorithms perform when applied feature selection to the performance evaluation of hepatocellular carcinoma[74]. Several classification techniques, including Support Vector Classification, Neural Network Classification, Random Forest, Logistic Regression, and Naive Bayes, will be combined with the Random Forest feature selection method. As an evaluation strategy, this study used 5-fold cross-validation. The findings revealed that the Random Forest algorithm, Neural Network, Vector Machine Classification, and Naive Bayes all exhibit higher classification performance evaluations than without the use of random forest feature selection, while the Logistic Regression model offers a higher performance evaluation without the use of random forest feature selection[74]. The highest performance evaluation is given to Support Vector Classification when compared to four other algorithms that use feature selection, however Logistic Regression delivers a superior performance evaluation when compared to various classification techniques that do not use feature selection.

2.3.57 A Comparison of Random Forest Variable Selection Methods for Classification Prediction Modeling

In this work, 311 freely accessible online datasets were used to compare the various strategies for random forest variable selection in the context of classification[75]. The

approaches with the smallest out-of-bag error rate, computation time, and variable count were preferred. High AUC values were only of tangential importance. Jiang and VSURF's approaches had the lowest overall error rate and best parsimony (fewest number of variables), but they also required longer computation durations. The Jiang approach employed k-fold validation to pick variables, which can be computationally expensive, which is one of the reasons why it took a long time to compute. The stepwise technique used by the VSURF method for picking variables, in which variables are eliminated and then perhaps reintroduced back, may be linked to a longer calculation time, is likely the source of the method's high computation times.

2.4 Summary of Literature Review

This study reviewed existing literatures to understand predictive models, cyberattacks, web attacks, machine learning techniques and algorithms. The reviews show that making use of data science, data mining, and machine learning has been directed towards the development for predictive models for different types of cyberattacks including web-based attack by several researchers. It also shows the different dataset and classification algorithm, mostly used for predictive models and their merit and demerit. Furthermore, the researchers also called for further researches on the subjects. This study will make use of a fraction of CIC-Bell-IDS2017 dataset that contains only web attack to create a predictive model specifically for those types of attacks and comparison of performance evaluation of random forest with and without feature selection, instead of making use of a general attack dataset like most other study.

Endnotes

[1] A. Tekerek, *"A Novel Architecture for Web-Based Attack Detection Using Convolution Neural Network,"* **Computers & Security**, 2020, pp 1-19, doi: https://doi.org/10.1016/j.cose.2020.102096

[2] D. Scheinost, S. Noble, C. Horien, A. S. Greene, E. Lake, M. Salehi, S. Gao, X. Shen, D. Oconnor, D. S. Barron, S. W. Yip, M. D. Rosenberg 7 R. T. Constable, *"Ten simple rules for predictive modeling of individual differences in neuroimaging,"* **Journal of Science Direct NeuroImage** 193, 2019, pp 35-45 https://doi.org/10.1016/j.neuroimage.2019.02.057

[3] S. Yusif & A. Hafeez-Baig, *"A Conceptual Model for Cybersecurity Governance,"* **Journal of Applied Security Research**, 2021, pp 1-25, DOI: 10.1080/19361610.2021.1918995

[4] J. J. Jeong ,J. Mihelcic, G. Oliver & C. Rudolp, *"Towards an Improved Understanding of Human Factors in Cybersecurity,"* **IEEE 5th International Conference on Collaboration and Internet Computing (CIC),** 2020, pp. 338-345 DOI 10.1109/CIC48465.2019.00047

[5] N. Agarwal & S. Z. Hussain, *"A Closer Look at Intrusion Detection System for Web Applications,"* **Hindawi Security and Communication Networks,** 2018, pp 1-27 https://doi.org/10.1155/2018/9601357

[6] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand & H. Liu, *"Advancing Feature Selection Research,"* **NSF-0812551,** 2017, pp 1-29 https://www.researchgate.net/publication/305083748

[7] A. Al-Zahrani, *"Assessing and Proposing Countermeasures for Cyber-Security Attacks,"* **International Journal of Advanced Computer Science and Applications (IJACSA)** 13, No 1, 2022, pp 885-895 www.ijacsa.thesai.org

[8] P. Angelo, A. Resende & A. C. Drummond, *"A Survey of Random Forest Based Methods for Intrusion Detection System,"* **ACM Computer Surveys** 51, 3, 2018, pp 1-36 https://doi.org/10.1145/3178582

[9] L. Schoppa, M. Disse & S. Bachmair, *"Evaluating the Performance of Random Forest for large-scale flood discharge simulation,"* **Journal of Hydrology Science Direct 590, GFZ German Research Centre for Geosciences, Germany & Institute of Environmental Science and Geography, University of Potsdam, Germany**, 2020, pp 125531 https://doi.org/10.1016/j.jhydrol.2020.125531

[10] R. Zuech, J. Hancock & T. M. Khoshgoftaar, *"Detecting Web Attacks Using Random Undersampling and Ensemble Learners,"* **Journal of Big Data**, 2021, pp 1-20 https://doi.org/10.1186/s40537-021-00460-8

[11] S. Leelavathy, R. Jaichandran, R. Shobana, S. Bhaskaran & A. Prathyunnan, *"A Secure Methodology to Detect and Prevent Ddos and Sql Injection Attacks,"* **Turkish Journal of Computer and Mathematics Education** Vol.12, No.2, 2021, pp 341- 346

[12] A. Gaurav, D. Santeniello, A. K. Gupta & F. Colace, *"A Biblometric review of the Current State and Future Perspectives of XSS attack detection in Web based Applications,"* **Preprint,** 2022, pp 1-12, DOI: 10.13140/RG.2.2.19829.65763

[13] I. Agrafiotis, M. Goldsmith, S. Creese & D. Upton, *"A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate,"* **Journal of Cybersecurity,** 2018, pp 1–15 doi: 10.1093/cybsec/tyy006

[14] M. S. Hasan & M. Nosonovsky, *"Triboinformatics: machine learning algorithms and data topology methods for tribology,"* **Surface Innovations** 10(4–5), 2022, pp 229–242, https://doi.org/10.1680/jsuin.22.00027

[15] I. H. Sarker, *"Machine Learning: Algorithms, Real-World Applications and Research Directions,"* **SN Computer Science** 160, 2021, pp 1-21 https://doi.org/10.1007/s42979-021-00592-x

[16] K. Akram, G. Banu, M. Basthikodi & A. R. Faizabadi, *"Defense Mechanism Using Multilayered Approach and SQL Injection Methods for Web Based Attacks,"* **Journal of Emerging Technologies and Innovative Research (JETIR)** Volume 6, Issue 5, 2019, pp 122-129

[17] D. Truong, D. Tran, L. Nguyen, H. Mac, H. A. Tran & T. Bui, *"Detecting Web Attacks using Stacked Denoising Autoencoder and Ensemble Learning Methods,"* **In The Tenth International Symposium on Information and Communication,** 2019, pp 1-6

[18] Y. Pan, F. Sun, Z. Teng, J. White, D. C. Schmidt, J. Staples & L. Krause, *"Detecting web attacks with end-to-end deep learning,"* **Journal of Internet Services and Applications** 10:16, 2019, pp 1-22 https://doi.org/10.1186/s13174-019-0115-x

[19] R. A. Katole, S. S. Sherekar & V. M. Thakare, *"Detection of SQL Injection Attacks by Removing the Parameter Values of SQL Query,"* **Proceedings of the Second International Conference on Inventive Systems and Control (ICISC),** 2018, pp 736-741

[20] M. H. AL-Maliki & M. N. Jasim, *"Review of SQL injection attacks: Detection, to enhance the security of the website from client-side attacks,"* **Int. J. Nonlinear Anal. Appl**. 2022, pp 3773-3782 http://dx.doi.org/10.22075/ijnaa.2022.6152

[21] M. Alsaffar, S. Aljaloud, B. A. Mohammed, Z. G. Al-Mekhlafi, T. S. Almurayziq, G. Alshammari & A. Alshammari, *"Detection of Web Cross-Site Scripting (XSS) Attacks,"* **Journal of Electronics**, 2022, pp 1-13 https:// doi.org/10.3390/electronics11142212

[22] S. Chavan & S. Tamane, *"Enhancement in Cloud Security for Web Application Attacks,"* **IEEE Xplore**, 2021, pp 91-95

[23] B. Kapoor & B. Nagpal, *"Ensemble Modelling for Predicting the Relation between Biopsychosocial Signals and Seizures using the Gradient Boosting Method,"* **Research Square**, 2022, pp 1-17 DOI: https://doi.org/10.21203/rs.3.rs-1810072/v1

[24] S. Sharma, P. Zavarsky & S. Butakov, *"Machine Learning based Intrusion Detection System for Web-Based Attacks,"* **IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS),** 2020, pp 1-4 DOI 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00048

[25] A. K. Keshri, A. Sharma, A. Chowdhury, S. S. Rawat & K. Kiran, *"SQL – Attacks, Modes, Prevention,"* **International Journal of Research in Engineering, Science and Management** Volume 5, Issue 1, 2022, pp 162-165

[26] G. Battineni, G. G. Sagaro, N. Chinatalapudi & F. Amenta, *"Applications of Machine Learning Predictive Models in the Chronic Disease Diagnosis,"* **Journal of Personalized Medicine,** volume 10, issue 21, 2020, pp 1-11 doi:10.3390/jpm10020021

[27] A. M. Vartouni, S. S. Kashi & M. Teshnehlab, *"An Anomaly Detection Method to Detect Web Attacks Using Stacked Auto-Encoder,"* **6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS),** 2018, pp 131-134

[28] A. N. Nazarov, D. V. Pantiukhin, I. M. Voronkov & M. A. Nazarov, *"Approach To Intelligent Monitoring Of Cyber Attacks,"* **Synchroinfo Journal** No. 6, 2020, pp 1-8 DOI: 10.36724/2664-066X-2020-6-6-2-9

[29] J. Diaz-Verdejo, J. Munoz-Calle, A. E. Alonso, R. E. Alonso & G. Madinabeitia, *"On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks,"* **Applied Sciences**, 12, 852, 2022, pp 1-16 http://doi.org/10.3390/app12020852

[30] M. Indushree, M. Kaur, R. Manish, R. Shashihara & L. Heung-No, *"Cross Channel Scripting and Code Injection Attacks on Web and Cloud-Based Applications: A Comprehensive Review,"* **Journal of Sensors**, 22, 1959, 2022, pp 1-20 https://doi.org/10.3390/s22051959

[31] A. Abdullah, Muhammad & M. Malik, *"A Survey on SQL Injection Attacks: Detection and Prevention,"* **Research Article,** 2022, pp 1-7 https://www.researchgate.net/publication/361444044

[32] E. Chatzoglou, G. Kambourakis & C. Kolias, *"Your WAP Is at Risk: A Vulnerability Analysis on Wireless Access Point Web-Based Management Interfaces,"* **Hindawi Security and Communication Networks,** 2022, pp 24 https://doi.org/10.1155/2022/1833062

[33] P. Sinha, A. Kumarrai & B. Bhushan, *"Information Security threats and attacks with conceivable counteraction,"* **2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT),** 2019, pp 1208-1213

[34] M. Husak, J. Komarkova, E. Bou-Harb & P. Celeda, *"Survey of Attack Projection, Prediction, and Forecasting in Cyber Security,"* **IEEE Communications Surveys & Tutorials**, 2018, pp 1-22, DOI 10.1109/COMST.2018.2871866

[35] T. S. Riera, J. B. Higuera, J. M. Herraiz & J. S. Montalvo, *"Prevention and Fighting against Web Attacks through Anomaly Detection Technology. A Systematic Review,"* **Journal of Sustainability,** 12, 4945, 2020, pp 1-45 doi:10.3390/su12124945

[36] A. Muhammad, M. Asad & A. R. Javed, *"Robust Early Stage Botnet Detection using Machine Learning,"* **IEEE Xplore**, 2021, pp 1-6 https://orcid.org/0000-0002-0570-1813

[37] R. SaiSindhuTheja & G. K. Shyam, *"An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment,"* **Applied Soft Computing Journal 100**, 106997, 2021, pp 1-11 https://doi.org/10.1016/j.asoc.2020.106997

[38] M. Shafiq, Z. Tian & A.K. Bashir, *"IoT malicious traffic identification using wrapper-based feature selection mechanisms,"* **Journal of Computers & Security 94**, 101863, 2020, pp 1-11 https://doi.org/10.1016/j.cose.2020.101863

[39] R. Utaya Surian, N. A. AbdRahman & Y. Nathan, *"Nscanner: Vulnerabilities Detection Tool for Web Application,"* **Journal of Physics: Conference Series 1712**, 012018, 2020, pp 1-10 doi:10.1088/1742-6596/1712/1/012018

[40] Y. Zhou, G. Cheng, S. Jiang & M. Dai, *"Building an efficient intrusion detection system based on feature selection and ensemble classifier,"* **Journal of Computer Networks** 174 ,107247, 2020, pp 1-17 https://doi.org/10.1016/j.comnet.2020.107247

[41] M. O. Miah, S. S. Khan, S. Shatabda & D. M. Farid, *"Improving Detection Accuracy for Imbalanced Network Intrusion Classification using Cluster-based Under-sampling with Random Forests,"* **1st International Conference on Advances in Science, Engineering and Robotics Technology** 2019, pp 1-5

[42] F. Salo, M. Injadat, A. Moubayed, A. B. Nassifi & A. Essex, *"Clustering Enabled Classification using Ensemble Feature Selection for Intrusion Detection,"* **Workshop on Computing, Networking and Communication (CNC**), 2019, pp 1-6

[43] F. S. L. Filho, F. A. F. Silveria, A. M. B. Junior, G. Vargas-Solar & L. F. Silveira, *"Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning."* **Journal of Security and Communication Networks,** 2019, pp 1-15https://doi.org/10.1155/2019/1574749

[44] J. Alsamiri & K. Alsubhi, *"Internet of Things Cyber Attacks Detection using Machine Learning,"* **(IJACSA) International Journal of Advanced Computer Science and Applications,** Vol. 10, No. 12, 2019, pp 627-634

[45] M. H. Kamarudin, C. Maple & T. Watson, *"Hybrid feature selection technique for intrusion detection system,"* **Int. J. High Performance Computing and Networking,** Vol. 13, No. 2, 2019, pp 232–240

[46] Y. Wan, J. Chang, R. Chen & S. Wang, *"Feature-Selection-Based Ransomware Detection with Machine Learning of Data Analysis,"* **3rd International Conference on Computer and Communication Systems,** 2018, pp 85-88

[47] S. Ustebay, Z. Turgut & M. A. Aydin, *"Intrusion Detection System with Recursive Feature Elimination by using Random Forest and Deep Learning Classifier,"* **International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism,** 2018, pp 71-76

[48] E. Semastin, S. Azam, B. Shanmugam, K. Kannoorpatti, M. Jonokmman, G. N. Samy & S. Perumal, *"Preventive Measures for Cross Site Request Forgery Attacks on Web-based Applications,"* **International Journal of Engineering & Technology,** 2018, pp 130-134

[49] G. Betarte, R. Martinez & A. Pardo, *"Web Application Attacks Detection Using Machine Learning Techniques,"* **17th IEEE International Conference on Machine Learning and Applications,** 2018, pp 1065-1072 DOI 10.1109/ICMLA.2018.00174

[50] C. J. Ugochukwu & E. O Bennett, *"An Intrusion Detection System Using Machine Learning Algorithm,"* **International Journal of Computer Science and Mathematical Theory** Vol. 4, No.1, 2018, pp 39-47

[51] A. Pattawaro & C. Polprasert, *"Anomaly-Based Network Intrusion Detection System through Feature Selection and Hybrid Machine Learning Technique,"* **Sixteenth International Conference on ICT and Knowledge Engineering,** 2018, pp 1-6

[52] R. C. Silva, M. P. O. Camargo, M. S. Quessada, A. C. Lopes, J. D. M. Ernesto, K. A. Pontara da Costa, *"An Intrusion Detection System for Web-Based Attacks Using IBM Watson,"* **Journal of IEEE Latin America Transactions** Volume: 20, Issue: 2, 2022, pp 191 – 197 DOI: 10.1109/TLA.2022.9661457

[53] A. A. Ashlam, A. Badii & F. Stahl, "*A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks,"* **5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET) in Hammamet, Tunisia,** 2022, DOI: 10.1109/IC_ASET53395.2022.9765948

[54] S. J. Y. Weamie, *"Cross-Site Scripting Attacks and Defensive Techniques: A Comprehensive Survey,"* **International Journal of Communications, Network and System Sciences** 15, 2022, pp 126-148 DOI: 10.4236/ijcns.2022.158010

[55] F. M. M. Mokbal, D. Wang & X. Wang, *"Detect Cross-Site Scripting Attacks Using Average Word Embedding and Support Vector Machine,"* **International Journal of Network Security,** Vol.24, No.1, 2022, pp 20-28 (DOI: 10.6633/IJNS.202201 24(1).03)

[56] X. Kuang, M. Zhang, H. Li, G. Zhao, Z. Wu & X. Wang, *"DeepWAF: Detecting Web Attacks Based on CNN and LSTM Models,"* **International Symposium on Cyberspace Safety and Security** (LNSC,volume 11983), 2019, pp 121–136

[57] X. Liu, Q. Yu, X. Zhou & Q. Zhou, *"OwlEye: An Advanced Detection System of Web Attacks Based on HMM,"* **IEEE 16th Intl Conf on Dependable, Autonomic and Secure**

Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech) Athens, Greece, 2018, 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00044

[58] O. E. Taylor & P. S. Ezekiel, "A Robust System for Detecting and Preventing Payloads Attacks on Web-Applications Using Recurrent Neural Network (RNN)," **European Journal of Computer Science and Information Technology,** 10 (4), 2022, pp 1-13

[59] V. S. Stency & N. Mohanasundaram, *"A Study on XSS Attacks: Intelligent Detection Methods,"* **Journal of Physics: Conference Series** 1767, 2021, pp 1-10 doi:10.1088/1742-6596/1767/1/012047

[60] B. Appiah, Z. Qin, O. A. Kwabena & M. A. Abdullah, *"Robust Training for Injection Attacks Detection in Web-based Applications,"* **International Journal of Network Security,** Vol.23, No.6, 2021, pp 1028-1036 (DOI: 10.6633/IJNS.202111 23(6).09)

[61] M. U. John, J. L. Shah & G. I. Ahmad, *"Web Abuse Using Cross Site Scripting (XSS) Attacks,"* **Journal of Artificial Intelligence Research & Advances**, Volume 6, Issue 1, 2019, pp 69-75

[62] S. Ray, *"A Quick Review of Machine Learning Algorithms,"* **International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con),** 2019, pp 1-5

[63] N. A. D. Suhaimi & H. Abas, *"A Systematic Literature Review On Supervised Machine Learning Algorithms,"* **PERINTIS eJournal**, Vol. 10, No. 1, 2020, pp 1-24

[64] B. Mahesh, *"Machine Learning Algorithms - A Review*," **International Journal of Science and Research (IJSR),** Volume 9 Issue 1, 2020, pp 381-386 DOI: 10.21275/ART2020399

[65] L. Yiheng & C. Weidong, *'' A Comparative Performance Assessment of Ensemble Learning for Credit Scorin, ''* **Journal of Mathematics**, 2020, pp 1-19

[66] A. Ons, P. Kandaraj, & P. Benoit, *'' Performance Evaluation of Feature Selection and Tree-Based Algorithms for Traffic Classification, ''* **PREPRINTS eJournal,** 2021, pp 1-7

[67] P. I. Maria, M. Nurulfa, & S. Kridanto, *'' Feature Selection to increase the Random Forest Method on High Dimensional Data, ''* **International Journal od Advances in Intelligent Informatics,** Vol. 9, No. 3, 2020, pp 303-312.

[68] M. A. Muhammad, *'' A Comparative Analysis of Classification Algorithm on Diverse Datasets, ''* **Engineering, Technology & Applied Science Research,** Vol. 8, No. 2, 2018, pp 2790-2795.

[69] S. Thanagapriya, & G. J. Nancy, *''A Comparison of Feature Selection Algorithms for Human Activity Recognition,''* **International Journal of Creative Research Thoughts,** Vol. 10, Issue 6, 2022, pp 640-647.

[70] S. T. Madhuri, & A. C. Amol, *'' Comparative Study of Supervised Algorithms for Prediction of Students Performance, ''* **International Journal of Modern Education and Computer Science,** 2021, pp 1-21.

[71] T. Arzina, S. Md, R.A. Mohammad, A. Jesmin, & R. M. AbusayedMd, *'' Performance Evaluation of Multiple Classifiers for Predicting Fake News, ''* **Journal of Computer and Communications,** 2022, 10, pp 1-21.

[72] K. M. Lubna, B. Mustafa, F. R. Ahmed, T. Ayshathul, & B. Safina, *''Evaluation of Correlation Feature Selection and Random Forest for Network Intrusion Detection, ''* **Journal of Emerging Technologies and Innovative Research,** Vol.6, Issue 5, 2019, pp 84-88

[73] K. M. Lubna, B. Mustafa, F. R. Ahmed, T. Ayshathul, & B. Safina, *'' Evaluation of Correlation Feature Selection and Random Forest for Network Intrusion Detection, ''* **Journal of Emerging Technologies and Innovative Research,** Vol.6, Issue 5, 2019, pp 84-88

[74] M. Abdul Latief, B. Alhadi, S. Titin, & S. Devvi, *''A Comparative Performance Evaluation of Random Forest Feature Selection on Classification of Hepatocellular Carcinoma Gene Expression Data, ''* **3rd International Conference on Informatics and Computational Sciences,** 2019, pp 1-6

[75] S. L. Jaime, M. E. Micheal, T. Janet, & Edwardip, *'' A Comparison of Random Forest Variable Selection Methods for Classification Prediction Modeling, ''* **Expert Syst. Appl.,** 2019, pp 1-17

Chapter Three

Methodology

3.1 Overview

Research methodology is the study of how a specific research project is been carried out using some laid down techniques or approach. It can also be seen as the scientific study of how a research problem is solved.

This chapter explains the research techniques and method that is used to achieve the stated aim and objectives of the study. This study proposes an architecture that uses Machine Learning Techniques, (Random Forest) and Feature Selection Technique to identify and predict network intrusions. It also intends to elaborate on all the stages and phases involved in the development of a predictive model for web-based attacks using random forest and correlation-based feature selection which includes planning, organizing and building up of every stage require to make the system model functional.

The step by step guide taking cognizance of the research aims and objectives is summarized in a work process diagram/conceptual framework/system architecture in Figure 3.1.

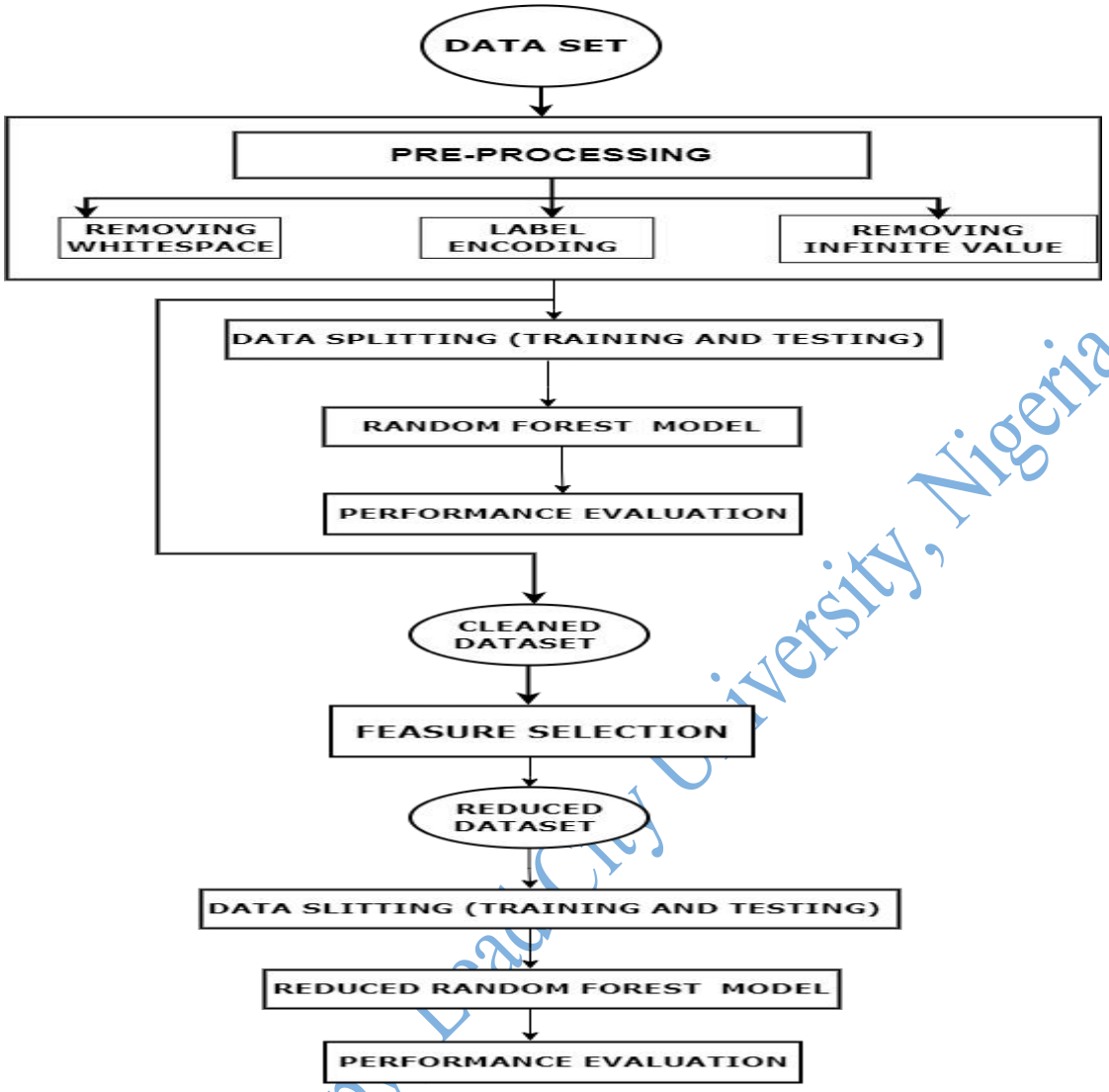


Figure 3.1: Conceptual Framework/ System Architecture

System architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system.

3.2 System Design

3.2.1 Stage 1: Data Collection and Description of Dataset

This research work used a fraction of a modern big intrusion dataset created by the Canadian Institute of Cybersecurity (CIC) and Bell Canada (BC) Cyber Threat Intelligence (CTI), generally refer to as the CIC-Bell-IDS2017 dataset, which contains common and modern attacks. The dataset was released in 2017 and available at: https://www.unb.ca/cic/datasets/ids2017.html The CIC-Bell-IDS2017 dataset comprises the most recent benign and common assaults. Additionally, it contains the outcomes of the CICFlowMeter network traffic analysis with flows categorized according to the time stamp, source and destination IP addresses, source and destination ports, protocols, and assaults (CSV files). This is one of the most recent intrusion detection datasets, and it contain current assaults including DDoS, Brute Force, Port Scan, Botnet, web attacks such as the XSS, SQL Injection, and Infiltration attacks. This dataset specifically consists of 2,830,743 records created on 8 files, each of which has 78 unique attributes and a label[2].

For this study the fraction of this dataset which contains 170,366 records of benign and common web attacks such as Cross-site scripting (XSS), Brute Force and SQL Injection is used to train and test the machine learning model for predicting web-based attacks.

Table 3.1: Distribution of Attacks

S/N	ATTACK TYPE	COUNT	DATA TYPE
0	Benign (Good Ware)	149113	Int64
1	Denial of Service (Dos Hulk)	22,380	Int64
2	Denial of Service (Golden Eye)	10,293	Int64
3	Distributed Denial of Service (Ddos)	20,317	Int64
4	Port Scan	11,002	Int64
5	Ftp-Patator	7,938	Int64
6	Ssh-Patator	5,897	Int64
7	Brute Force (Web Attack)	1,470	Int64
8	Cross Site Scripting (Web Attack)	652	Int64
9	Infiltration	36	Int64
10	Sql Injection (Web Attack)	21	Int64
11	Heartbleed	11	Int64

3.2.2 Stage 2: Pre-Processing / Data Cleaning

The dataset is cleaned by the data preprocessing module, and provides adequate data for anomaly detection. The original data's default values and infinity values are processed in this

module[1].Prior to training, data is pre-processed and normalized. This is the process of removing irrelevant or superfluous information from data and keep only the most crucial and significant information[3]. The most time-consuming and crucial phase in data mining is data pretreatment. Realistic data can be noisy, redundant, partial, and inconsistent and is frequently derived from diverse platforms. To make raw data appropriate for analysis and knowledge discovery, it is crucial to alter it[2]. Most big dataset contains large, redundant, semi-structured, and unstructured data that present challenges to knowledge discovery and data modeling. Data cleaning is the act of preparing data for analysis by removing, cleaning, or modifying data that are out of order. Therefore, the preprocessing/data cleaning stage in this study includes Dropping Unwanted Column, Replacing NaN value or Whitespace, removing infinite value, and Label Encoding (for binary classification where all attacks are grouped as "Attack" and good-ware as "Benign).

3.2.2.1 Dropping Unwanted Column

This is also known as eliminating unneeded columns. Unwanted columns are characteristics that don't improve data modeling. Because it has no link to the output value and can lead the model to make poor decisions, the destination port that contained undesirable socket information was removed from the study's set of characteristics.

3.2.2.2 Removing infinite value and replacing nan value or whitespace

In this phase, infinite values are eliminated, and NaN values are substituted. The dataset's infinite values were deleted, and all NaN values were changed to zero prior to the creation of the prediction models (0). This is done to make sure that every cell has a value because NaN and infinite values have a negative impact on the models' ability to learn from datasets,

causing the classifiers to reject them. Infinite values are values that have no terminator or are longer than the permitted data type, while whitespaces are cells that contain NaN values or empty attributes or features (float32).

3.2.2.3 Label Encoding

Label encoding is a popular technique for handling categorical variable. It is the act of grouping and encoding multiple classes into two possible class or category. This study employed label encoding for binary classification, therefore all attack types in the dataset are labelled as "Attack" and all good-ware as "Benign", and encoded as 0.0 and 1.0, where 0.0 represents Benign and 1.0 represent Attacks.

3.2.3 Stage 3: Data Splitting for Raw Dataset

This is the process of dividing data into two, where a certain percentage is used for training the ML algorithm 80% training and 20% for testing

3.2.4 Stage 4: Development of Predictive Model for Clean Dataset

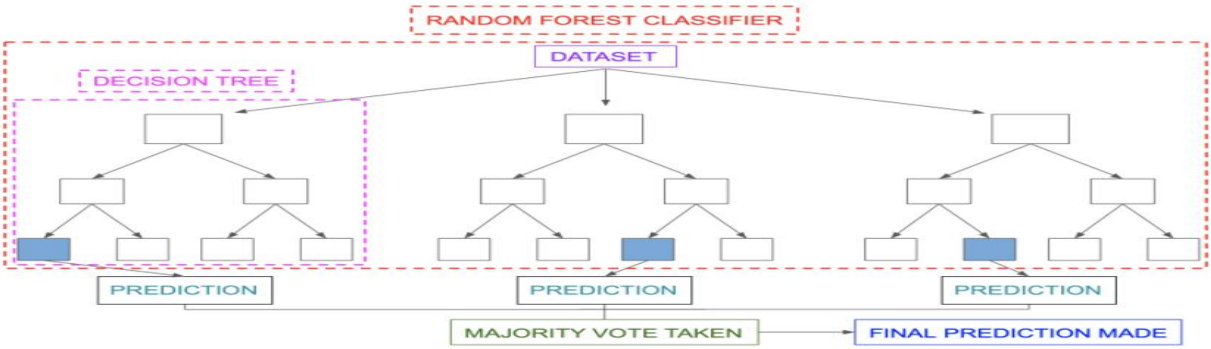
At this stage of this work Random Forest that operates by constructing a multiple Decision Trees, is used on the cleaned dataset to generate a predictive model, using data balancing and 80:20 split tests. The models will be evaluated using confusion matrix to determine the accuracy, precision, recall, and F-1 score of each model.

Description of Machine Learning Algorithms: Random Forest

Random Forest can be thought of as an ensemble of classification trees, where each tree casts one vote for the task of determining the most common class from the input data[2]. Decision trees are used in the machine learning method known as Random Forest. This approach

assembles a large number of different decision tree structures that are formed in various ways to form a "forest"[6]. The efficiency with which it can process large datasets, its small weight in comparison to other approaches, and its robustness against noise and outliers as compared to single classifiers are just a few of the benefits of this technique[6].

It is made up of a number of decision trees, and random forest combines them together to acquire precise estimation and produce more accurate outcomes. Random forests are wonderful because they can be applied to both classification and regression. Additionally, Random Forest provides us with guidance on which important features should be kept and which ones should be dropped from the dataset[5]. An assembly of independent decision trees is called Random Forest (RF). Every individual decision tree first categorizes each instance, and the instance is then finally classified by the collective wisdom of all the individual trees[8]. RF is an ensemble classifier that is built from a variety of different decision tree classifiers. To put it another way, ensemble classifiers are more complex because they combine different independent classifiers. In several instances, it has been demonstrated that ensembles outperform their non-ensemble counterparts. Additionally, ensembles have become common in Kaggle tournaments[8]. Decision trees display the predictions that come from a succession of feature-based splits using a flowchart that resembles a tree structure. The decision is made by the leaves at the end, which follows the root node[7]. With a decision tree as a base, Random-forest does both row sampling and column sampling.



Source: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

Using this method, the Gini index heuristics are used to evaluate the subset of characteristics chosen in each interior node. The Gini index is a function that assesses the impureness of data and event uncertainty. By calculating the Gini of each branch on a node using class and probability, this method can predict which branch is most likely to occur. The formula for GINI's general form is:

Gini (t) = 1- $\sum_{i=1}^{N} P(C_i|t)^2$ or Gini = 1- $\sum_{i=1}^{N} (P_i)^2$

Another way to describe this is the use of entropy to determine how nodes branch in a decision tree based on the probability of a specific outcome.

$$\text{Entropy} = \sum_{i=1}^{N} - P_i * log_2 (P_i)$$

- P and P_i represents the relative frequency of the class.
- T is a condition,
- N the number of classes in the data set, and
- C_i is the i^{th} class label in the data set.

Entropy examines the likelihood of a particular result to determine which branch the node should take. It is more mathematically complex than the Gini index since a logarithmic function is utilized to calculate it[8].

3.2.5 Stage 5: Evaluation of the Raw Model

In this study the performance of the model is evaluated using Confusion Matrix.

Description of Evaluation Tool: Confusion Matrix

A confusion matrix, which using true and false detection of the model to determines the classification accuracy, precision, recall, and F-1 score. A common structure for evaluating accuracy is the confusion matrix, commonly referred to as the error matrix. It primarily serves as a means of contrasting classification outcomes with actual measured values. In a confusion matrix, it can show the classification findings' accuracy[1].

Table 3.2: Confusion matrix for binary classification

PARAMETER (REAL LIFE)	ATTACK	BENIGN
Attack	True Positive	False Negative
Benign	False Negative	True Negative

Source:https://www.sciencedirect.com/topics/engineering/confusion-matrix#:~:text=A%20confusion%20matrix%20is%20a,performance%20of%20a%20classification%20algorithm.

- Where True Positive (TP) is the correctly predicted Attack
- True Negative (TN) is the correctly predicted Benign.
- False Negative (FN) is Attack that failed to be identified or predicted as Benign.
- False Positive (FP) are Benign that failed to be identified or predicted as Attack

Accuracy determines the percentage of correctly classified instances. Out of all the samples in the dataset, it is the proportion of properly predicted samples. Given as;

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Precision gives the percentage of true positive instance that are correctly classified. Finding the probability that a positive forecast will come true requires precision. Given as;

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Recall is used to calculate the model's ability to predict positive value. Given as;

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

F-measure is defined as the harmonic mean of Precision and Recall (when both considered)

$$F\text{-}Measure = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

3.2.6 Stage 6: Feature Selection

The process of evaluating each feature individually to decide which ones within the dataset have the most impact on the outcome is known as feature selection[7]. The goal is to reduce the dimensions of high dimensional data while maintaining the same accuracy, if not higher.

At this stage, correlation-based feature selection will be used for feature selection, in order to improve the model performance. Feature selection is the process of automatically selecting or

extracting the most relevant and valuable features using mathematics, statistics, and domain knowledge. It is used to boost the accuracy and minimize the error rate of machine learning models. The process can be sped up through feature selection, which can also assist in choosing the ideal machine learning strategy that will finally yield effective results[5].

The concept of correlation is used to compare two different features. For example, if the features are uncorrelated, the correlation will be zero; if not, it will be 1. To calculate the correlation between the two distinct variables, two complete modules—classical linear correlation and correlation on the basis of information theory—were put into use[9].

Correlation can be calculated via few methods

To calculate the correlation between two variable x and y, using Pearson correlation coefficient:

$$r = \frac{\Sigma[(x - \bar{x})(y - \bar{y})]}{\Sigma(x - \bar{x})^2 \, \Sigma(y - \bar{y})^2}$$

- Where r is the correlation coefficient
- x and y are the variables
- $\bar{x}$ is mean of x,
- $\bar{y}$ is the mean of y

NOTE:

- if r is between 0.6 and 1 then a Positive correlation exist
- if r is between -0.6 and -1 then a Negative correlation exist
- if r is 0 then no correlation whatsoever (Neutral)

- if r is closer to 0 than 1 (<=0.5) then weak correlation exit

Feature Scaling Techniques

Feature scaling also referred to as data normalization is an approach used to improve the learning process of ML models by scaling the input value. It is used to correct the error and compromises that comes with handling features that have similar and disproportionate scale, or feature with drastically different scale[8]. Data Normalization uses min-max scaling and standardization (zero score Normalization) to solve this problem.

i. Min-max scaling maps a numerical value x to the (0,1) interval

$$x^i = \frac{x - min}{\max - min} \qquad 3.1$$

ii. Standardization (also called Z-score normalization) maps a numerical value x to a new distribution with (mean) μ = 0 and (standard deviation) σ = 1

$$x^i = \frac{x - \mu}{\sigma} \qquad 3.2$$

Where min is the minimum value or data range

max is the maximum value or data range

x is the original value to scale

x^i is the normalized value

μ is the mean

σ is the standard deviation

3.2.7 Stage 7: Development of Predictive Model for Reduced Dataset

At this stage, the Random Forest machine learning classifier will be used again to train the reduced dataset (using the same approach in stage 4).

3.2.8 Stage 8: Evaluation of the Reduced Model

Finally the performance of the reduced model will also be evaluated using confusion matrix to determine the performance. This is expected to show improvement from the result generated in stage 5.

3.3 Requirement Specifications

The requirement specification to achieve the success of this system required both hardware and software tools. The hardware tools are those physical electronic devices while the software tools are the written instructions in forms of programs.

3.3.1 Hardware Implementation Tools

The experiments were done on a 64-bitWindows7 operating system with 4GB of RAM and a Intel Pentium (R) Dual core CPU at 1.90GHz per core.

3.3.2 Software Implementation Tools

Jupyter Notebook is a python web-based interactive graphic user programing interface and development environment for data mining, machine leaning and data analysis libraries and tools: PySpark, MLib, SkLearn, Tensorflow, Pandas, Numpy, and more. It provides a single comprehensive environment for data scientist to run many scientific packages that would have required many different individual programing environment and package.

Jupyter Notebook, which is an interactive Anaconda Navigator environment that helps scientists manipulate data, analyze data, and create machine learning model using Python Programming Language[10].

3.4 Research Methods

This research adopted a Constructive research methodology. The Constructive research method is mostly used in software engineering and computer science research by constructing diagrams, models, plans[9].

3.4.1 Data Collection Methods

Primary Data Collection Methods: Gathering information directly from an original source is one of the primary data collection methods. Data was gathered by the organization for use by specialists. Prior to consulting secondary or tertiary sources, primary data collecting entails acquiring information. It involves gathering information from a real-world source, such as a client or user. Both human and automatic methods are acceptable for doing this[9].

Secondary Data Collection Methods: On the other hand, secondary data collection methods refer to information gathered and made accessible to other researchers by a party other than the original user. Collecting data from databases or other sources, such public records, is what this procedure entails[10].

The methods of data collection adopted for this research work is secondary data collection method which is the dataset created by the Canadian Institute of Cybersecurity (CIC) and Bell Canada (BC) Cyber Threat Intelligence (CTI).

Endnotes

[1] L. Xukui, C. Wei, Z Qianru, & W. Lifa, *''Building Auto-Encoder Intrusion Detection System Based on Random Forest Feature Selection,''* **Computer & Security**, 95 ,2020, pp 101851.

[2] Z.Yuyang, C. Guang, J. Shanqing, & D. Mian, *'' Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier, ''* **Computer Networks** 174, 2020, pp 107247.

[3] Kunal, & D. Mohit, *''Attribute Selection and Ensemble Classifier Based Novel Approach To Intrusion Detection System,''* **International Conference on Computational Intelligence and Data Science** (ICCIDS 2019), 167, 2020, pp 2191-2199.

[4] A. H. Faezah, A. L. Wathiq, & I. K. Ali, *''Differential Evolution Wrapper Feature Selection for Intrusion Detection System,''* **International Conference on Computational Intelligence and Data Science** (ICCIDS 2019), 167, 2020, pp 1230-1239.

[5] M. Ali, A. Muhammad, & J. R. Abdul, *''Robust Early Stage Botnet Detection using Machine Learning,''* **IEEE,** 978 , 2020, pp 1-7281-6840-1.

[6] A. Jadel, & A. Khalid, *''Internet of Things Cyber Attacks Detection using Machine Learning,''* **International Journal of Advanced Computer Science and Applications,** Vol. 10, No. 12, 2019.

[7] U. Serpil, T. Zeynep, & A. A. Muhammad, *''Intrusion Detection System with Recursive Feature Elimination by using Random Forest and Deep Learning Classifier, ''* **International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism,** 2018, 978-1-7281-0472-0.

[8] Z. Richard, H. John, & K. M. Taghi, *''Detecting Web Attacks using Random undersampling and Ensemble Learners, ''* **Journal of Big Data**, 2021, pp 8;75.

[9] H. Zulfiqar, L. Q. Huang, L. H. Sun, Y. F. Dao, & H. Lin, *''Deep-4Mcgp: A Deep Learning Approach to Predict 4mC sites in Geobacter Pickeringii by using Correlation-Based Feature Selection Techniques,''* **International Journal of Molecule Sciences**, 2022, pp 1251.

[10] K. H. Muhammad, M. Carsten, & W. Tim, *''Hybrid Feature Selection Technique for Intrusion Detection System, ''* **International Journal of Performance Computing and Networking,** Vol. 13, No. 2, 2019.

Chapter Four

Result and Discussion of Findings

This chapter discusses the experiments done in line with the proposed architecture and approach explained in the chapter three of this study. It shows the system implementation and evaluation of results.

4.2 Predictive Model Development

The predictive models were developed with python programming language, using Jupyter Notebook, which is a python-based programming environment and interface for Anaconda Navigator. Anaconda Navigator is a desktop graphical user interface (GUI) that allows users to launch applications and easily manage conda packages, libraries, and environments without using command-line interface. Conda is a package and environment management framework that helps data scientist and programmers to find and install many data science libraries required for machine learning tasks and data analysis. It provides a single comprehensive environment for data scientist to run many scientific packages that would have required many different individual programing environment and package. Among this are environment like Jupyter notebook, JupyterLab, PyCharm, Glueviz, Orange 3, RStudio and VSCode (anaconda.com). Anaconda navigator was installed and used in this study to provide access to machine learning and data analytics libraries needed, and also to provide access to Jupyter Notebook environment, which is a python web-based interactive graphic user programing interface and development environment for notebooks, code, and data. It allows codes and their corresponding output (result) to be displayed on a single web interface, referred to as notebook. It is a flexible interface that allows users to configure and arrange

workflows in data science, scientific computing and machine learning (jupyternotebook.com). Jupyter Notebook is used in this study as the major Python application programming interface and to incorporate machine learning and data analytics frameworks, libraries and toolkit: PySpark, MLLib, SkLearn, Tensorflow, Pandas, and Numpy within a single python programming interface. Python is an object-oriented programming language. This means that any program can be solved in python by creating a model. However, python cab be treated as procedural as well as structural language.

4.3 Experimental Results

This section shows the performance of Random Forest (RF) for the task of detecting web attack. In the experiments the fraction of CIC-IDS2017 dataset described in chapter three was balanced and used to train and test the model. The model generated used 80% training set and 20% testing set, before and after feature selection. The result of the two different models are shown in Table 4.1 for the raw dataset, and Table 4.2 for the reduced dataset.

4.3.1 Performance Evaluation without Feature Selection for the Raw Dataset

This section shows the results of the performance analysis (Accuracy, Attack Precision, Attack Recall, Model Development time and Attack F-1 score of Random Forest (RF), on the raw dataset.

Table 4.1: Random Forest (RF) without Feature Selection for the Raw Dataset

Performance Matrix	Prediction Score
Precision	97%
Recall	96%
F1-Score	98%
Classification Accuracy	94%
Model Development time	35 sec

4.3.2 Performance Evaluation with Feature Selection of the Dataset

This section shows the results of the performance analysis (Accuracy, Precision, Recall, and F-1 score) of Random Forest (RF), for the reduced dataset.

Table 4.2: Random Forest (RF) with Feature Selection of the Reduced Dataset

Performance Matrix	Prediction Score
Precision	98%
Recall	97%
F1-Score	99%
Classification Accuracy	99%
Model Development time	15 sec

Table 4.3: Comparative Analysis of the Performance Evaluation for both Raw and Reduced Dataset.

CLASIFIERS	Classification Accuracy	Precision	Recall	F1-Score	Model Development Time
Raw set	94%	97%	96%	98%	35 sec
Reduced set	99%	98%	97%	99%	15 sec

4.4 Discussion of Results

From the results of the experiment carried out, it was discovered that there were improvements in the performance of the Random forest classifier, in all categories of evaluation: Accuracy, Precision, Recall, F1-score, and Model Development Time, after correlation technique was used for feature selection, except for precision that produce 100% for both full and reduced set. A good recall was also generated for both the raw and the reduced dataset, achieving 96% and 97% respectively. The result also produced a great precision and model development time in both cases, with precision of 97% and 98% for both set, and model development time of 35sec for raw set and 15sec for reduced set, which is a vital point in deploying machine learning model in a real-time environment. Furthermore, even with the ensemble learning ability of the Random Forest classifier, the experimental result shows an increase of 5% in the classification accuracy after feature selection, which is a shred of evidence that even the much-celebrated ensemble learning algorithm can be improved with feature selection. Other areas of comparison between the two approaches include the quantity of features chosen, the length of training and testing, bias toward particular characteristics, and other performance metric A comparison shows that even though Random Forest has higher predictive accuracy than CFS based classifier, it is

computationally expensive. Both of these techniques are suitable with their own merits and demerits.

Chapter Five

Conclusion

5.1 Conclusion

As a result of traditional security systems' repeated failures to identify complex and novel attacks, the security industry has recently come under harsh criticism. However, the Anomaly Intrusion Detection System (AIDS) that employs machine learning methodology is an efficient tool in identifying these attacks and comparing the performance evaluation of Random Forest with and without Feature Selection, as demonstrated in this research. This study has demonstrated how the predictive power of machine learning models can be increased by using the big data analytics tool (PySpark). The findings of the classification algorithms Random Forest for both raw and reduced dataset were examined in terms of their appropriateness for identifying intrusions from a sizable dataset comprising numerous contemporary attacks using the Python programming language. The use of big data analytics (PySpark) was found to help machine learning models perform better, resulting in a better intrusion detection system. Researchers can compare the classification accuracy with other approaches, such as SVM, NN, and NN with K fold, deep learning, and possibly improve the accuracy of the classification.

5.2 Recommendation and Future Works

It's vital to realize that each of these algorithms has its own advantages and disadvantages as well as a unique knowledge base and method of approaching the issue. Machine learning is an experimental science that has been used to perform the task of detecting intrusion. As a result, a learning technique or classifier that excels at one problem may not necessarily

perform well at another. Likewise, a classifier that generates good performance measurements for one dataset may produce poor performance measurements for another dataset or a different set of features.

Based on the aforementioned information, the remaining work on this thesis can be summed up as follows:

1. Researchers can compare the classification accuracy with other approaches, such as SVM, NN, and NN with K fold, deep learning, and possibly improve the accuracy of the classification.
2. The dataset may benefit from the adoption of alternative classification algorithms, which also results in models with improved performances.
3. A portion of the actual dataset was used in this study due to the limited processing speed and memory made available for the tests that were done. The CIS-IDS2017 dataset can be used for training in these trials, which could help the model perform better if distributed processing using Apache Spark is used.
4. Correlation analysis served as the foundation for the feature selection method used in this thesis. It is feasible to run tests in which a new method will be applied, most likely choosing various attributes.

5.3 Contribution to Knowledge

This study added to the body of knowledge in the fields of Machine Learning, Big Data Analytics, and Feature Selection. It also evaluated the performance of random forests with and without feature selection in comparison and showed how these fields can be combined to create a viable intrusion detection system. It also demonstrates how big data analytics may be

used to select and scale features, enhancing or improving the performance of conventional machine learning models for intrusion detection.

Bibliography

Conference Paper

Acar G., Huang D.Y., Li F., Narayanan A., & Feamster N., *"Web Based Attacks to Discover and Control Local IoT Devices,"* In IoT S&P: ACM SIGCOMM, 2018, pp 29-35 https://doi.org/10.1145/3229565.3229568

Jeong J.J., Mihelcic J., Oliver J., & Rudolp C., *"Towards an Improved Understanding of Human Factors in Cybersecurity,"* IEEE 5th International Conference on Collaboration and Internet Computing (CIC), 2020, pp 338-345 DOI 10.1109/CIC48465.2019.00047

Katole R.A., Sherekar S.S., & Thakare V.M., *"Detection of SQL Injection Attacks by Removing the Parameter Values of SQL Query,"* Proceedings of the Second International Conference on Inventive Systems and Control (ICISC), 2018, pp 736-741

Chavan S., & Tamane S., *"Enhancement in Cloud Security for Web Application Attacks,"* IEEE Xplore, 2021, pp 91-95

Sharma S., Zavarsky P., & Butakov S., *"Machine Learning based Intrusion Detection System for Web-Based Attacks,"* IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), July 2020, pp 1-4 DOI 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00048

Vartouni A.M., Kashi S.S., & Teshnehlab M., *"An Anomaly Detection Method to Detect Web Attacks Using Stacked Auto-Encoder,"* 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), 2018, pp 131-134

Sinha P., Kumarrai A. & Bhushan B., *"Information Security threats and attacks with conceivable counteraction,"* 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2019, pp 1208-1213

Husak M., Komarkova J., Bou-Harb E., & Celeda P., *"Survey of Attack Projection, Prediction, and Forecasting in Cyber Security,"* IEEE Communications Surveys & Tutorials, 2018, pp 1-22, DOI 10.1109/COMST.2018.2871866

Muhammad A., Asad M., & Javed A.R., *"Robust Early Stage Botnet Detection using Machine Learning,"* IEEE Xplore, 2021, pp 1-6 https://orcid.org/0000-0002-0570-1813

Miah M.O., Khan S.S., Shatabda S., & Farid D.M., *"Improving Detection Accuracy for Imbalanced Network Intrusion Classification using Cluster-based Under-sampling*

with Random Forests," 1st International Conference on Advances in Science, Engineering and Robotics Technology, 2019, pp 1-5

Salo F., Injadat M., Moubayed A., Nassifi A.B., & Essex A., *"Clustering Enabled Classification using Ensemble Feature Selection for Intrusion Detection,"* Workshop on Computing, Networking and Communication (CNC), 2019, pp 1-6

Wan Y., Chang J., Chen R., & Wang S., *"Feature-Selection-Based Ransomware Detection with Machine Learning of Data Analysis,"* 3rd International Conference on Computer and Communication Systems, 2018, pp 85-88

Ustebay S., Turgut Z., & Aydin M.A.,*"Intrusion Detection System with Recursive Feature Elimination by using Random Forest and Deep Learning Classifier,"* International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, 2018, pp 71-76

Betarte G., Martinez R., & Pardo A.,*"Web Application Attacks Detection Using Machine Learning Techniques,"* 17th IEEE International Conference on Machine Learning and Applications, 2018, pp 1065-1072 DOI 10.1109/ICMLA.2018.00174

Pattawaro A., & Polprasert C., *"Anomaly-Based Network Intrusion Detection System through Feature Selection and Hybrid Machine Learning Technique,"* Sixteenth International Conference on ICT and Knowledge Engineering, 2018, pp 1-6

Ashlam A.A., Badii A., & Stahl F., "*A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks,"* 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET) in Hammamet, Tunisia, 2022, DOI: 10.1109/IC_ASET53395.2022.9765948

Kuang X., Zhang M., Li H., Zhao G., Wu Z., & Wang X., *"DeepWAF: Detecting Web Attacks Based on CNN and LSTM Models,"* International Symposium on Cyberspace Safety and Security (LNSC,volume 11983), 2019, pp 121–136

Liu X., Yu Q., Zhou X., & Zhou Q., *"OwlEye: An Advanced Detection System of Web Attacks Based on HMM,"* IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech) Athens, Greece, 2018, 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00044

Ray S., *"A Quick Review of Machine Learning Algorithms,"* International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con**),** 2019, pp 1-5

Kunal, & Mohit D, *''Attribute Selection and Ensemble Classifier Based Novel Approach To Intrusion Detection System,''* International Conference on Computational Intelligence and Data Science (ICCIDS 2019), 167, 2020, pp 2191-2199.

Faezah A.H., Wathiq A.L., & Ali I.K., *''Differential Evolution Wrapper Feature Selection for Intrusion Detection System,''* International Conference on Computational Intelligence and Data Science (ICCIDS 2019), 167, 2020, pp 1230-1239.

Ali M., Muhammad A., & Abdul J.R., *''Robust Early Stage Botnet Detection using Machine Learning,''* IEEE, 978, 2020, 1-7281-6840-1.

Serpil U., Zeynep T., & Muhammad A.A., *'' Intrusion Detection System with Recursive Feature Elimination by using Random Forest and Deep Learning Classifier, ''* International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, 2018, 978-1-7281-0472-0.

Journal

Singh A., Kumar A., & Bharti A.K., "*Identification and Prevention approaches for Web-based Attacks using Machine Learning Techniques,"* **International Journal of Creative Research Thoughts (IJCRT)** 2, vol. 9, 2021, pp 4558-4563.

Riera T.S., Higuera J.B., Herraiz J.M., & Montalvo J.S., *"A New Multi-label Dataset for Web Attacks CAPEC Classification using Machine Learning Techniques,"* **Journal of Science Direct Computer and Security** 120, 2022, pp 1-18

Sharif M.H., *"Web Attacks Analysis and Mitigation Techniques,"* **International Journal of Engineering Research and Technology (IJERT),** 2022, www.ijert.org

Zwilling M., Klien G., Lesjak D., Wiechetek L., Cetin F., & Basim H.M., *"CyberSecurity Awareness, Knowledge and Behavior: A Comparative Study,"* **Journal of Computer Information System,** 2020, pp 1-16

Agarwal N., & Hussain S.Z., *"A Closer Look at Intrusion Detection System for Web Applications,"* **Hindawi Security and Communication Networks,** 2018, pp 27 https://doi.org/10.1155/2018/9601357

Karuparthi B., & Mahesh A., *"Comparative Study between Random Forest and Support Vector Machine Algorithm in Classifying Cervical Cancer,"* **International Journal of Engineering Research & Technology (IJERT),** Vol. 11 Issue 01, 2022, pp 434-437

Li W., *"Optimization and Application of Random Forest Algorithm for Applied Mathematics Specialty,"* **Hindawi Security and Communication Networks,** 2022, pp 9 https://doi.org/10.1155/2022/1131994

Singh S., Choudhary S.S., & Bhavishya S., *"Feature Selection Effects on Classification Algorithms: Laconic description of Machine Learning Algorithms,"* **International**

Journal of Engineering Research & Technology (IJERT), Vol. 7 Issue 02, 2018, pp 183-185 http://www.ijert.org

Al-Shalabi L., *"New Feature Selection Algorithm Based on Feature Stability and Correlation,"* **Journal of Information Technology and Computing,** 2017, pp 1-16 DOI 10.1109/ACCESS.2022.3140209, IEEE Access

Avkurova Z., *"Models for Early Web – Attacks Detection and Intruders Identification Based on Fuzzy Logic,"* **Journal of Science Direct: Procedia Computer Scienec** 198, 2022, pp 694-699.

Avkurova Z., Gnatyuk S., Abduraimova B., Fedushko S., Syerov Y., & Trach O., *"Detecting web Attacks using Random Under sampling and Ensemble Learners,"* **Journal of Big Data**, 2021, pp 1-20 https://doi.org/10.1186/s40537-021-00460-8

Tekerek A., *"A Novel Architecture for Web-Based Attack Detection Using Convolution Neural Network,"* **Journal of Computers & Security,** 2020, pp 1-19, doi: https://doi.org/10.1016/j.cose.2020.102096

Scheinost D., Noble S., Horien C., Greene A.S., Lake E., Salehi M., Gao S., Shen X., Oconnor D., Barron D.S., Yip S.W., Rosenberg M.D., & Constable R.T., *"Ten simple rules for predictive modeling of individual differences in neuroimaging,"* **Journal of Science Direct NeuroImage** 193, 2019, pp 35-45 https://doi.org/10.1016/j.neuroimage.2019.02.057

Yusif S., & Hafeez-Baig A., *"A Conceptual Model for Cybersecurity Governance,"* **Journal of Applied Security Research**, 2021, pp 1-25, DOI: 10.1080/19361610.2021.1918995

Al-Zahrani A., *"Assessing and Proposing Countermeasures for Cyber-Security Attacks,"* **International Journal of Advanced Computer Science and Applications (IJACSA)** 13, No 1, 2022, pp 885-895 www.ijacsa.thesai.org

Schoppa L., Disse M., & Bachmair S., *"Evaluating the Performance of Random Forest for large-scale flood discharge simulation,"* **Journal of Hydrology Science Direct 590, GFZ German Research Centre for Geosciences, Germany & Institute of Environmental Science and Geography, University of Potsdam, Germany**, 2020, pp 125531 https://doi.org/10.1016/j.jhydrol.2020.125531

Zuech R., Hancock J., & Khoshgoftaar T.M., *"Detecting Web Attacks Using Random Undersampling and Ensemble Learners,"* **Journal of Big Data**, 2021, pp 1-20 https://doi.org/10.1186/s40537-021-00460-8

Leelavathy S., Jaichandran R., Shobana R., Bhaskaran S., & Prathyunnan A., *"A Secure Methodology to Detect and Prevent Ddos and Sql Injection Attacks,"* **Turkish Journal of Computer and Mathematics Education** Vol.12, No.2, 2021, pp 341-346

Agrafiotis I., Goldsmith M.,, Creese S., & Upton D.,, *"A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate,"* **Journal of Cybersecurity,** 2018, pp 1–15 doi: 10.1093/cybsec/tyy006

Akram K., Banu G., Basthikodi M., & Faizabadi A.R., *"Defense Mechanism Using Multilayered Approach and SQL Injection Methods for Web Based Attacks,"* **Journal of Emerging Technologies and Innovative Research (JETIR)** Volume 6, Issue 5, 2019, pp 122-129

Truong D., Tran D., Nguyen L., Mac H., Tran H.A., & Bui T., *"Detecting Web Attacks using Stacked Denoising Autoencoder and Ensemble Learning Methods,"* **In The Tenth International Symposium on Information and Communication,** 2019, pp 1-6

Pan Y., Sun F., Teng Z., White J., Schmidt D.C., Staples J., & Krause L., *"Detecting web attacks with end-to-end deep learning,"* **Journal of Internet Services and Applications** 10:16, 2019, pp 1-22 https://doi.org/10.1186/s13174-019-0115-x

AL-Maliki M.H., & Jasim M.N., *"Review of SQL injection attacks: Detection, to enhance the security of the website from client-side attacks,"* **Int. J. Nonlinear Anal. Appl**. 13,1 ,2022, pp 3773-3782 http://dx.doi.org/10.22075/ijnaa.2022.6152

Alsaffar M., Aljaloud S., Mohammed B.A., Al-Mekhlafi Z.G., Almurayziq T.S., Alshammari G., & Alshammari A., *"Detection of Web Cross-Site Scripting (XSS) Attacks,"* **Journal of Electronics** 11, 2212, 2022, pp 1-13 https:// doi.org/10.3390/electronics11142212

Keshri A.K., Sharma A., Chowdhury A., Rawat S.S., & Kiran K., *"SQL – Attacks, Modes, Prevention,"* **International Journal of Research in Engineering, Science and Management** Volume 5, Issue 1, 2022, pp 162-165

Battineni G., Sagaro G.G., Chinatalapudi N., & Amenta F., *"Applications of Machine Learning Predictive Models in the Chronic Disease Diagnosis,"* **Journal of Personalized Medicine** 10, 21, 2020, pp 1-11 doi:10.3390/jpm10020021

Nazarov A.N., Pantiukhin D.V., Voronkov I.M., & Nazarov M.A., *"Approach To Intelligent Monitoring Of Cyber Attacks,"* **Synchroinfo Journal** No. 6, 2020, pp 1-8 DOI: 10.36724/2664-066X-2020-6-6-2-9

Indushree M., Kaur M., Manish R., Shashihara R., & Heung-No L., *"Cross Channel Scripting and Code Injection Attacks on Web and Cloud-Based Applications: A Comprehensive Review,"* **Journal of Sensors**, 22, 1959, 2022, pp 1-20 https://doi.org/10.3390/s22051959

Chatzoglou E., Kambourakis G., & Kolias C., *"Your WAP Is at Risk: A Vulnerability Analysis on Wireless Access Point Web-Based Management Interfaces,"* **Hindawi Security and Communication Networks,** 2022, pp 24 pages https://doi.org/10.1155/2022/1833062

Riera T.S., Higuera J.B., Herraiz J.M., & Montalvo J.S., *"Prevention and Fighting against Web Attacks through Anomaly Detection Technology. A Systematic Review,"* **Journal of Sustainability,** 12, 4945, 2020, pp 1-45 doi:10.3390/su12124945

SaiSindhuTheja R., & Shyam G.K., *"An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment,"* **Applied Soft Computing Journal 100**, 106997, 2021, pp 1-11 https://doi.org/10.1016/j.asoc.2020.106997

Shafiq M., Tian Z., & Bashir A.K., *"IoT malicious traffic identification using wrapper-based feature selection mechanisms,"* **Journal of Computers & Security 94**, 101863, 2020, pp 1-11 https://doi.org/10.1016/j.cose.2020.101863

Utaya Surian R., AbdRahman N.A., & Nathan Y., *"Nscanner: Vulnerabilities Detection Tool for Web Application,"* **Journal of Physics: Conference Series 1712**, 012018, 2020, pp 1-10 doi:10.1088/1742-6596/1712/1/012018

Zhou Y., Cheng G., Jiang S., & Dai M., *"Building an efficient intrusion detection system based on feature selection and ensemble classifier,"* **Journal of Computer Networks** 174 ,107247, 2020, pp 1-17 https://doi.org/10.1016/j.comnet.2020.107247

Filho F.S.L., Silveria F.A.F., Junior A.M.B., Vargas-Solar G., & Silveira L.F., *"Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning,"* **Journal of Security and Communication Networks,** 2019, pp 1-15https://doi.org/10.1155/2019/1574749

Alsamiri J., & Alsubhi K., *"Internet of Things Cyber Attacks Detection using Machine Learning,"* **(IJACSA) International Journal of Advanced Computer Science and Applications,** Vol. 10, No. 12, 2019, pp 627-634

Kamarudin M.H., Maple C., & Watson T.,, *"Hybrid feature selection technique for intrusion detection system,"* **Int. J. High Performance Computing and Networking,** Vol. 13, No. 2, 2019, pp 232–240

Semastin E., Azam S., Shanmugam B., Kannoorpatti K., Jonokmman M., Samy G.N., & Perumal S., *"Preventive Measures for Cross Site Request Forgery Attacks on Web-based Applications,"* **International Journal of Engineering & Technology,** 7 (4.15), 2018, pp 130-134

Ugochukwu C.J., & Bennett E.O., *"An Intrusion Detection System Using Machine Learning Algorithm,"* **International Journal of Computer Science and Mathematical Theory** Vol. 4, No.1, 2018, pp 39-47

Silva R.C., Camargo M.P.O., Quessada M.S., Lopes A.C., Ernesto J.D.M., & Pontara da Costa K.A., *"An Intrusion Detection System for Web-Based Attacks Using IBM Watson,"* **Journal of IEEE Latin America Transactions** Volume: 20, Issue: 2, 2022, pp 191 – 197 DOI: 10.1109/TLA.2022.9661457

Weamie S.J.Y., *"Cross-Site Scripting Attacks and Defensive Techniques: A Comprehensive Survey,"* **International Journal of Communications, Network and System Sciences** 15, 2022, pp 126-148 DOI: 10.4236/ijcns.2022.158010

Mokbal F.M.M., Wang D., & Wang X., *"Detect Cross-Site Scripting Attacks Using Average Word Embedding and Support Vector Machine,"* **International Journal of Network Security,** Vol.24, No.1, 2022, pp 20-28 (DOI: 10.6633/IJNS.202201 24(1).03)

Taylor O.E., & Ezekiel P.S.,"A Robust System for Detecting and Preventing Payloads Attacks on Web-Applications Using Recurrent Neural Network (RNN)," **European Journal of Computer Science and Information Technology,** 10 (4), 2022, pp 1-13

Stency V.S., & Mohanasundaram N., *"A Study on XSS Attacks: Intelligent Detection Methods,"* **Journal of Physics: Conference Series** 1767, 2021, pp 1-10 doi:10.1088/1742-6596/1767/1/012047

Appiah B., Qin Z., Kwabena O.A., & Abdullah M.A., *"Robust Training for Injection Attacks Detection in Web-based Applications,"* **International Journal of Network Security,** Vol.23, No.6, 2021, pp 1028-1036 (DOI: 10.6633/IJNS.202111 23(6).09)

John M.U., Shah J.L., & Ahmad G.I., *"Web Abuse Using Cross Site Scripting (XSS) Attacks,"* **Journal of Artificial Intelligence Research & Advances**, Volume 6, Issue 1, 2019, pp 69-75

Jadel A., & Khalid A., *''Internet of Things Cyber Attacks Detection using Machine Learning,''* **International Journal of Advanced Computer Science and Applications,** Vol. 10, No. 12, 2019, pp 63-69

Richard Z., John H., & Taghi K.M., *''Detecting Web Attacks using Random undersampling and Ensemble Learners,''* **Journal of Big Data**, 2021, pp 8-75.

Zulfiqar H., Huang L.Q., Sun L.H., Dao Y.F., & Lin H., *''Deep-4Mcgp: A Deep Learning Approach to Predict 4mC sites in Geobacter Pickeringii by using Correlation-Based Feature Selection Techniques,''* **International Journal of Molecule Sciences**, 2022, pp 23, 1251.

Muhammad K.H., Carsten M., & Tim W, *''Hybrid Feature Selection Technique for Intrusion Detection System,''* **International Journal of Performance Computing and Networking,** Vol. 13, No. 2, 2019.

Research Articles

Chowdhurry R., Banerjee P., Deep Dey S., Saha B., & Bandyopadhyay S.K., *"A Decision Tree Based Intrusion Detection System For Identification of Malicious Web Based Attacks,"* Preprints (www.preprints.org), vol. 1 2020.

Agarwal N., & Hussain S.Z., *"A Closer Look at Intrusion Detection System for Web Applications,"* Hindawi Security and Communication Networks, 2018, pp 1-27 https://doi.org/10.1155/2018/9601357

Zhao Z., Morstatter F., Sharma S., Alelyani S., Anand A. & Liu H., *"Advancing Feature Selection Research,"* NSF-0812551, 2017, pp 1-29 https://www.researchgate.net/publication/305083748

Angelo P., Resende A., & Drummond A.C., *"A Survey of Random Forest Based Methods for Intrusion Detection System,"* ACM Computer Surveys 51, 3, 2018, pp 1-36 https://doi.org/10.1145/3178582

Gaurav A., Santeniello D., Gupta A.K., & Colace F., *"A Biblometric review of the Current State and Future Perspectives of XSS attack detection in Web based Applications,"* Preprint May 2022, pp 1-12, DOI: 10.13140/RG.2.2.19829.65763

Hasan M.S., & Nosonovsky M., *"Triboinformatics: machine learning algorithms and data topology methods for tribology,"* Surface Innovations 10(4–5), 2022, pp 229–242, https://doi.org/10.1680/jsuin.22.00027

Sarker I.H., *"Machine Learning: Algorithms, Real-World Applications and Research Directions,"* SN Computer Science 160, 2021, pp 1-21 https://doi.org/10.1007/s42979-021-00592-x

Kapoor B., & Nagpal B., *"Ensemble Modelling for Predicting the Relation between Biopsychosocial Signals and Seizures using the Gradient Boosting Method,"* Research Square 1, 2022, pp 1-17 DOI: https://doi.org/10.21203/rs.3.rs-1810072/v1

Diaz-Verdejo J., Munoz-Calle J., Alonso A.E., Alonso R.E., & Madinabeitia G., *"On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks,"* Applied Sciences, 12, 852, 2022, pp 1-16 http://doi.org/10.3390/app12020852

Abdullah A., Muhammad, & Malik M., *"A Survey on SQL Injection Attacks: Detection and Prevention,"* Research Article, 2022, pp 1-7 https://www.researchgate.net/publication/361444044

Suhaimi N.A., & Abas H., *"A Systematic Literature Review On Supervised Machine Learning Algorithms,"* PERINTIS eJournal, Vol. 10, No. 1, 2020, pp 1-24

Xukui L., Wei C., Qianru Z., & Lifa W., *''Building Auto-Encoder Intrusion Detection System Based on Random Forest Feature Selection,"* Computer & Security, 95 ,2020, pp 101851.

Yuyang Z., Guang C., Shanqing J., & Mian D., *'' Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier, ''* Computer Networks 174, 2020, pp 107247.

Appendix

Appendix A: Initial Data Input and Modeling

Observations

The labels are imbalanced. The dataset contains 78 features. For the initial modeling with raw data, IP ad ID columns were dropped. The column names had trailing spaces that affected data subsettiing with column names, this was resolved using the pandas str.strip() method. The dataset column names contains a lot of messy data, model fitting with raw data is impossible

due to value errors.

In [1]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
from datetime import datetime
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

In [2]:
```
initial_df = pd.read_csv('MachineLearningCVE/Thursday-WorkingHours-Morning-WebAttack
```

In [3]:
```
initial_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>

RangeIndex: 170366 entries, 0 to 170365

Data columns (total 79 columns):

 #   Column                       Non-Null Count   Dtype

---  ------                       --------------   -----
 0    Destination Port            170366 non-null  int64
 1    Flow Duration               170366 non-null  int64
 2    Total Fwd Packets           170366 non-null  int64
 3    Total Backward Packets      170366 non-null  int64
 4   Total Length of Fwd Packets  170366 non-null  int64
 5    Total Length of Bwd Packets 170366 non-null  int64
 6    Fwd Packet Length Max       170366 non-null  int64
 7    Fwd Packet Length Min       170366 non-null  int64
 8    Fwd Packet Length Mean      170366 non-null  float64
 9    Fwd Packet Length Std       170366 non-null  float64
 10  Bwd Packet Length Max        170366 non-null  int64
 11   Bwd Packet Length Min       170366 non-null  int64
 12   Bwd Packet Length Mean      170366 non-null  float64
 13   Bwd Packet Length Std       170366 non-null  float64
 14  Flow Bytes/s                 170346 non-null  float64
 15   Flow Packets/s              170366 non-null  float64
 16   Flow IAT Mean               170366 non-null  float64
 17   Flow IAT Std                170366 non-null  float64
 18   Flow IAT Max                170366 non-null  int64
 19   Flow IAT Min                170366 non-null  int64
 20  Fwd IAT Total                170366 non-null  int64
 21   Fwd IAT Mean                170366 non-null  float64
 22   Fwd IAT Std                 170366 non-null  float64
 23   Fwd IAT Max                 170366 non-null  int64
 24   Fwd IAT Min                 170366 non-null  int64
 25  Bwd IAT Total                170366 non-null  int64
 26   Bwd IAT Mean                170366 non-null  float64
 27   Bwd IAT Std                 170366 non-null  float64
 28   Bwd IAT Max                 170366 non-null  int64
 29   Bwd IAT Min                 170366 non-null  int64
 30  Fwd PSH Flags                170366 non-null  int64
 31   Bwd PSH Flags               170366 non-null  int64
 32   Fwd URG Flags               170366 non-null  int64
 33   Bwd URG Flags               170366 non-null  int64
 34   Fwd Header Length           170366 non-null  int64
 35   Bwd Header Length           170366 non-null  int64
 36  Fwd Packets/s                170366 non-null  float64
 37   Bwd Packets/s               170366 non-null  float64
 38   Min Packet Length           170366 non-null  int64
 39   Max Packet Length           170366 non-null  int64
 40   Packet Length Mean          170366 non-null  float64
 41   Packet Length Std           170366 non-null  float64
 42   Packet Length Variance      170366 non-null  float64
 43  FIN Flag Count               170366 non-null  int64
```

```
 44   SYN Flag Count               170366 non-null  int64
 45   RST Flag Count               170366 non-null  int64
 46   PSH Flag Count               170366 non-null  int64
 47   ACK Flag Count               170366 non-null  int64
 48   URG Flag Count               170366 non-null  int64
 49   CWE Flag Count               170366 non-null  int64
 50   ECE Flag Count               170366 non-null  int64
 51   Down/Up Ratio                170366 non-null  int64
 52   Average Packet Size          170366 non-null  float64
 53   Avg Fwd Segment Size         170366 non-null  float64
 54   Avg Bwd Segment Size         170366 non-null  float64
 55   Fwd Header Length.1          170366 non-null  int64
 56  Fwd Avg Bytes/Bulk            170366 non-null  int64
 57   Fwd Avg Packets/Bulk         170366 non-null  int64
 58   Fwd Avg Bulk Rate            170366 non-null  int64
 59   Bwd Avg Bytes/Bulk           170366 non-null  int64
 60   Bwd Avg Packets/Bulk         170366 non-null  int64
 61  Bwd Avg Bulk Rate             170366 non-null  int64
 62  Subflow Fwd Packets           170366 non-null  int64
 63   Subflow Fwd Bytes            170366 non-null  int64
 64   Subflow Bwd Packets          170366 non-null  int64
 65   Subflow Bwd Bytes            170366 non-null  int64
 66  Init_Win_bytes_forward        170366 non-null  int64
 67   Init_Win_bytes_backward      170366 non-null  int64
 68   act_data_pkt_fwd             170366 non-null  int64
 69   min_seg_size_forward         170366 non-null  int64
 70  Active Mean                   170366 non-null  float64
 71   Active Std                   170366 non-null  float64
 72   Active Max                   170366 non-null  int64
 73   Active Min                   170366 non-null  int64
 74  Idle Mean                     170366 non-null  float64
 75   Idle Std                     170366 non-null  float64
 76   Idle Max                     170366 non-null  int64
 77   Idle Min                     170366 non-null  int64
 78   Label                        170366 non-null  object

dtypes: float64(24), int64(54), object(1)

memory usage: 102.7+ MB
```

Remove Spaces from Column names

In [4]:
```
initial_df.columns = initial_df.columns.str.strip()
```

In [5]:
```
initial_df.head(2)
```

Out[5]:

	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	P Le
0	389	113095465	48	24	9668	10012	403	0	201.416667	203.54
1	389	113473706	68	40	11364	12718	403	0	167.117647	171.91

2 rows × 79 columns

Drop unwanted Columns

In [6]:
```
columns_to_drop = ['Destination Port']

initial_df.drop(columns_to_drop, axis=1, inplace=True)
```

View target distribution

In [7]:
```
initial_df['Label'].value_counts()
```

Out[7]:
```
BENIGN                     168186
Web Attack  Brute Force      1507
Web Attack  XSS                652
Web Attack  Sql Injection       21
Name: Label, dtype: int64
```

Combine all type of Web attacks as Web attack

Since the distribution of target variables is so imbalanced, ml will perform better detecting web attack as a group than detecting individual attack types

In [8]:
```
initial_df['Label'] = initial_df['Label'].apply(lambda x: x if x == 'BENIGN' else 'W
```

In [9]:
```
initial_df['Label'].value_counts()
```

Out[9]:
```
BENIGN        168186
Web Attack      2180
Name: Label, dtype: int64
```

Get statistical distribution of dataset

In [10]:
```
initial_df.describe()
```

Out[10]:

	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet
count	1.703660e+05	170366.000000	1.703660e+05	170366.000000	1.703660e+05	170366.000

	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	
mean	1.246354e+07	15.124620	18.022276	5.569859e+02	3.183147e+04	167.775982	
std	3.193852e+07	1123.107756	1494.492871	7.710431e+03	3.460816e+06	461.299214	
min	-1.000000e+00	1.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	
25%	1.920000e+02	1.000000	1.000000	3.100000e+01	6.000000e+00	23.000000	
50%	3.141200e+04	2.000000	2.000000	6.800000e+01	1.340000e+02	41.000000	
75%	8.169818e+05	4.000000	2.000000	1.480000e+02	3.280000e+02	60.000000	
max	1.200000e+08	200755.000000	270686.000000	1.197199e+06	6.270000e+08	23360.000000	1

8 rows × 77 columns

Label Encoding

Observation: harmless network flows are labelled 0 and web attacks are labeled 1

In [11]:
```
from sklearn.preprocessing import OrdinalEncoder


ord_enc = OrdinalEncoder()
initial_df["Label_code"] = ord_enc.fit_transform(initial_df[["Label"]])
```

In [12]:
```
initial_df["Label_code"].value_counts()
```

Out[12]:
```
0.0    168186
1.0      2180
Name: Label_code, dtype: int64
```

1.2 Initial modeling on Raw data

Observation: model cannot fit to raw data as a result of presence of infinite values and values too large for float 32. More data processing to remove infinite values and scale values will be carried out

In [13]:
```
features = initial_df.columns.to_list()
features.remove('Label')
features.remove('Label_code')
```

In [14]:
```
X = initial_df[features]
Y = initial_df["Label_code"]
```

In [15]:
```
#set stratify=Y because the target values are not evenly distributed
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.2,random_state=0
```

In [16]:
```
#Fitting Random forest classification to training set
RFclassifier = RandomForestClassifier( n_estimators = 100,criterion= 'entropy',class
RFclassifier.fit(X_train, Y_train)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/_x/z410g4hn2k54q0l4wkv9dgxh0000gn/T/ipykernel_1177/1999772788.py in <mo
dule>
      1 #Fitting Random forest classification to training set
      2 RFclassifier = RandomForestClassifier( n_estimators = 100,criterion= 'entrop
y',class_weight='balanced_subsample')
----> 3 RFclassifier.fit(X_train, Y_train)

/opt/anaconda3/lib/python3.9/site-packages/sklearn/ensemble/_forest.py in fit(self,
X, y, sample_weight)
    302                 "sparse multilabel-indicator for y is not supported."
    303             )
--> 304         X, y = self._validate_data(X, y, multi_output=True,
    305                                    accept_sparse="csc", dtype=DTYPE)
    306         if sample_weight is not None:

/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py in _validate_data(self,
X, y, reset, validate_separately, **check_params)
    431                 y = check_array(y, **check_y_params)
    432             else:
--> 433                 X, y = check_X_y(X, y, **check_params)
    434             out = X, y
    435
```

```
/opt/anaconda3/lib/python3.9/site-packages/sklearn/utils/validation.py in inner_f(*a
rgs, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0


/opt/anaconda3/lib/python3.9/site-packages/sklearn/utils/validation.py in check_X_y
(X, y, accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ens
ure_2d, allow_nd, multi_output, ensure_min_samples, ensure_min_features, y_numeric,
estimator)
    869         raise ValueError("y cannot be None")
    870
--> 871     X = check_array(X, accept_sparse=accept_sparse,
    872                     accept_large_sparse=accept_large_sparse,
    873                     dtype=dtype, order=order, copy=copy,


/opt/anaconda3/lib/python3.9/site-packages/sklearn/utils/validation.py in inner_f(*a
rgs, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0


/opt/anaconda3/lib/python3.9/site-packages/sklearn/utils/validation.py in check_arra
y(array, accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, e
nsure_2d, allow_nd, ensure_min_samples, ensure_min_features, estimator)
    718
    719         if force_all_finite:
--> 720             _assert_all_finite(array,
    721                                allow_nan=force_all_finite == 'allow-nan')
    722


/opt/anaconda3/lib/python3.9/site-packages/sklearn/utils/validation.py in _assert_al
l_finite(X, allow_nan, msg_dtype)
    101                 not allow_nan and not np.isfinite(X).all()):
    102             type_err = 'infinity' if allow_nan else 'NaN, infinity'

--> 103             raise ValueError(
    104                     msg_err.format
    105                     (type_err,
```

```
ValueError: Input contains NaN, infinity or a value too large for dtype('float32').
```

Fix null and infinity values

In [17]:
```
# Replace infinity values with null
initial_df.replace([np.inf, -np.inf], np.nan, inplace=True)
# Fill null with
initial_df.fillna(0, inplace=True)
```

Export Cleaned Data for Next Stage

In [18]:
```
initial_df.to_csv('Cleaned.csv', index =False)
```

In [19]:
```
# delete initial df
del initial_df
```

In [20]:
```
start=datetime.now()

#Statements

print (datetime.now()-start)
```

```
0:00:00.000052
```

Appendix B: Exploratory Data Analysis, Data processing and Model Training

In this stage, the cleaned data is explored, corrrelation analysis is used for feature reduction because the dataset is quite large to avoid overfitting, reduce model run time and computational power required, normalization is carried out on all features, dataset is split and

model fitting, predictions and accuracy is carried out.

In [21]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt pd.set_option('display.max_colwidth',
None) pd.set_option('display.max_rows', None, 'display.max_columns',
None) import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Import data

In [22]:
```
df = pd.read_csv('Cleaned.csv')
```

In [23]:
```
df.head(4)
```

Out[23]:

	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	Bwd Packet Length Max
0	113095465	48	24	9668	10012	403	0	201.416667	203.548293	923
1	113473706	68	40	11364	12718	403	0	167.117647	171.919413	1139
2	119945515	150	0	0	0	0	0	0.000000	0.000000	0
3	60261928	9	7	2330	4221	1093	0	258.888889	409.702161	1460

In [24]:
```
df.shape
```

Out[24]:
```
(170366, 79)
```

In [25]:
```
df.dtypes
```

Out[25]:

Flow Duration	int64
Total Fwd Packets	int64
Total Backward Packets	int64
Total Length of Fwd Packets	int64
Total Length of Bwd Packets	int64
Fwd Packet Length Max	int64
Fwd Packet Length Min	int64
Fwd Packet Length Mean	float64
Fwd Packet Length Std	float64
Bwd Packet Length Max	int64
Bwd Packet Length Min	int64
Bwd Packet Length Mean	float64
Bwd Packet Length Std	float64
Flow Bytes/s	float64
Flow Packets/s	float64
Flow IAT Mean	float64
Flow IAT Std	float64
Flow IAT Max	int64
Flow IAT Min	int64
Fwd IAT Total	int64
Fwd IAT Mean	float64
Fwd IAT Std	float64
Fwd IAT Max	int64
Fwd IAT Min	int64
Bwd IAT Total	int64
Bwd IAT Mean	float64
Bwd IAT Std	float64
Bwd IAT Max	int64
Bwd IAT Min	int64

Fwd PSH Flags	int64
Bwd PSH Flags	int64
Fwd URG Flags	int64
Bwd URG Flags	int64
Fwd Header Length	int64
Bwd Header Length	int64
Fwd Packets/s	float64
Bwd Packets/s	float64
Min Packet Length	int64
Max Packet Length	int64
Packet Length Mean	float64
Packet Length Std	float64
Packet Length Variance	float64
FIN Flag Count	int64
SYN Flag Count	int64
RST Flag Count	int64
PSH Flag Count	int64
ACK Flag Count	int64
URG Flag Count	int64
CWE Flag Count	int64
ECE Flag Count	int64
Down/Up Ratio	int64
Average Packet Size	float64
Avg Fwd Segment Size	float64
Avg Bwd Segment Size	float64
Fwd Header Length.1	int64
Fwd Avg Bytes/Bulk	int64
Fwd Avg Packets/Bulk	int64
Fwd Avg Bulk Rate	int64
Bwd Avg Bytes/Bulk	int64
Bwd Avg Packets/Bulk	int64
Bwd Avg Bulk Rate	int64
Subflow Fwd Packets	int64
Subflow Fwd Bytes	int64

```
Subflow Bwd Packets              int64
Subflow Bwd Bytes                int64
Init_Win_bytes_forward           int64
Init_Win_bytes_backward            int64
act_data_pkt_fwd                   int64
min_seg_size_forward             int64
Active Mean                    float64
Active Std                     float64
Active Max                       int64
Active Min                       int64
Idle Mean                      float64
Idle Std                       float64
Idle Max                         int64
Idle Min                         int64
Label                             object
Label_code                       float64
dtype: object
```

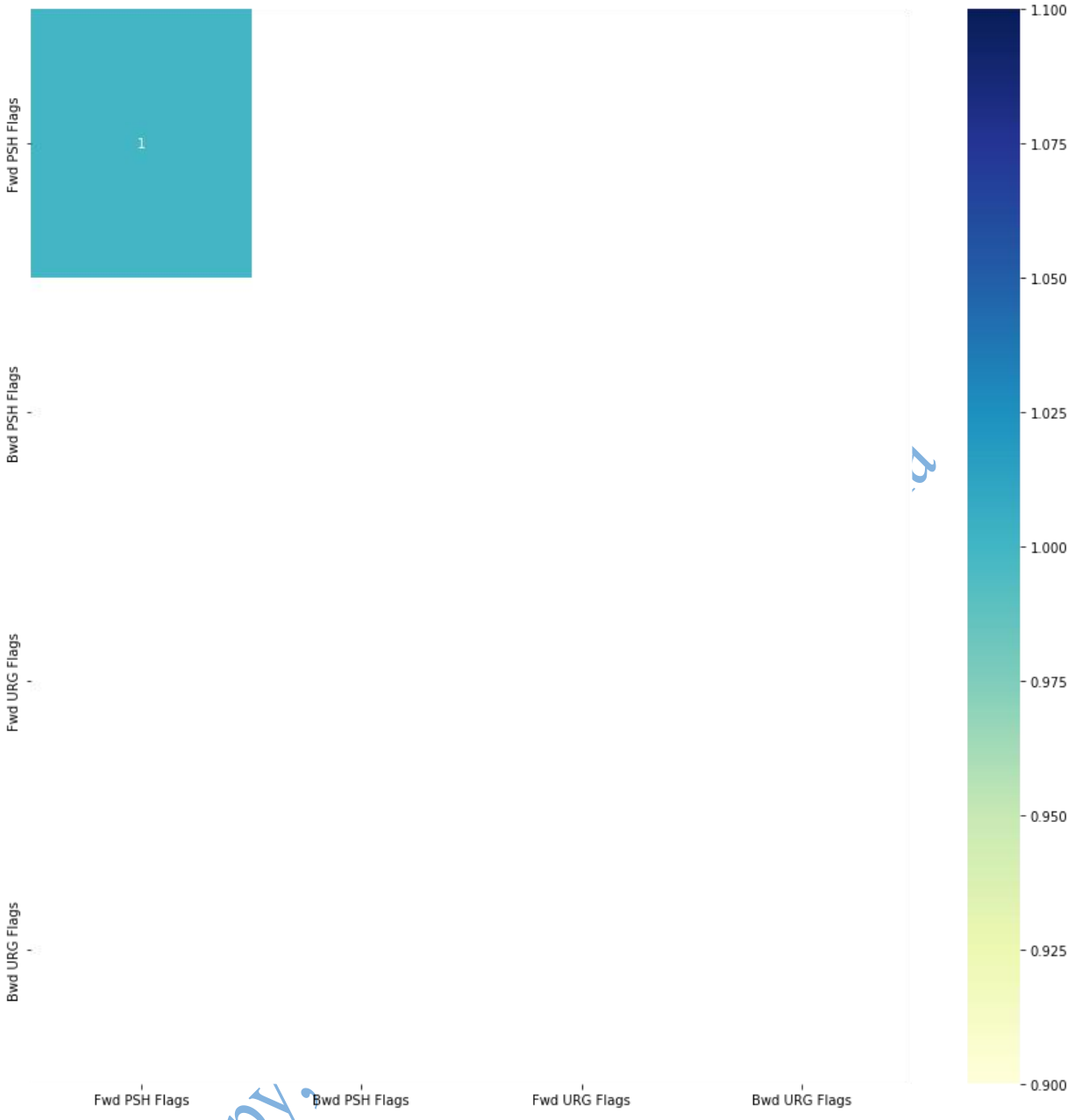
Appendix C: Correlation Analysis

In [26]:
```
plt.rcParams['figure.figsize'] = (15, 15)
```

In [27]:
```
features = df.columns.to_list()
```

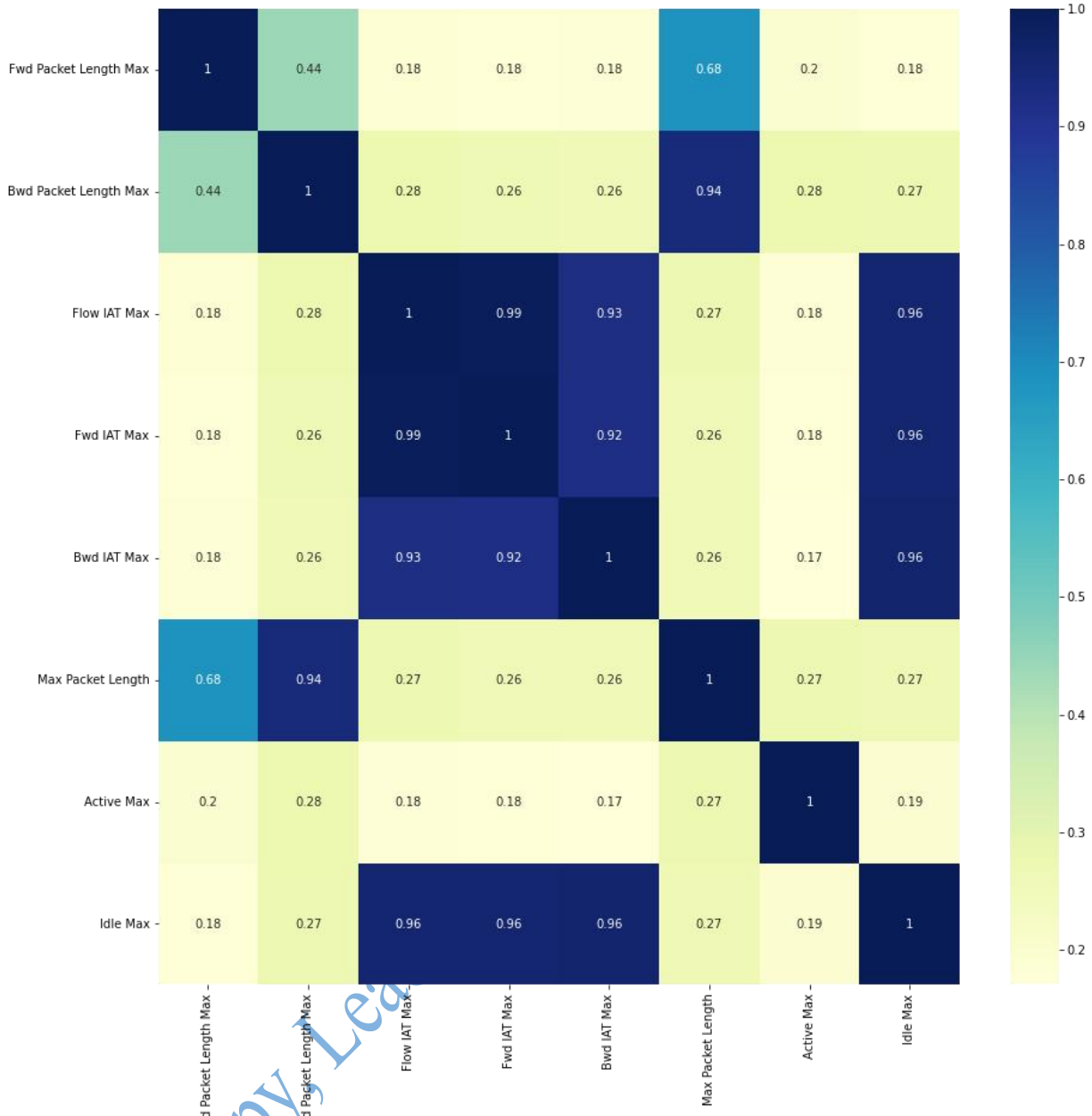
In [28]:
```
def plot_corr(cols: list,df=df):
    sns.heatmap( df[cols].corr(), cmap="YlGnBu", annot=True)
    plt.show()


def drop_cols(cols_to_drop: list):
    for col in cols_to_drop:
        try:
            features.remove(col)
        except ValueError:
            print(f'Unable to remove {col}')
```

In [29]:
```
# Check correlation in columns that have flags in them
cols = [x for x in df.columns if 'Flags' in x]

plot_corr(cols)
```



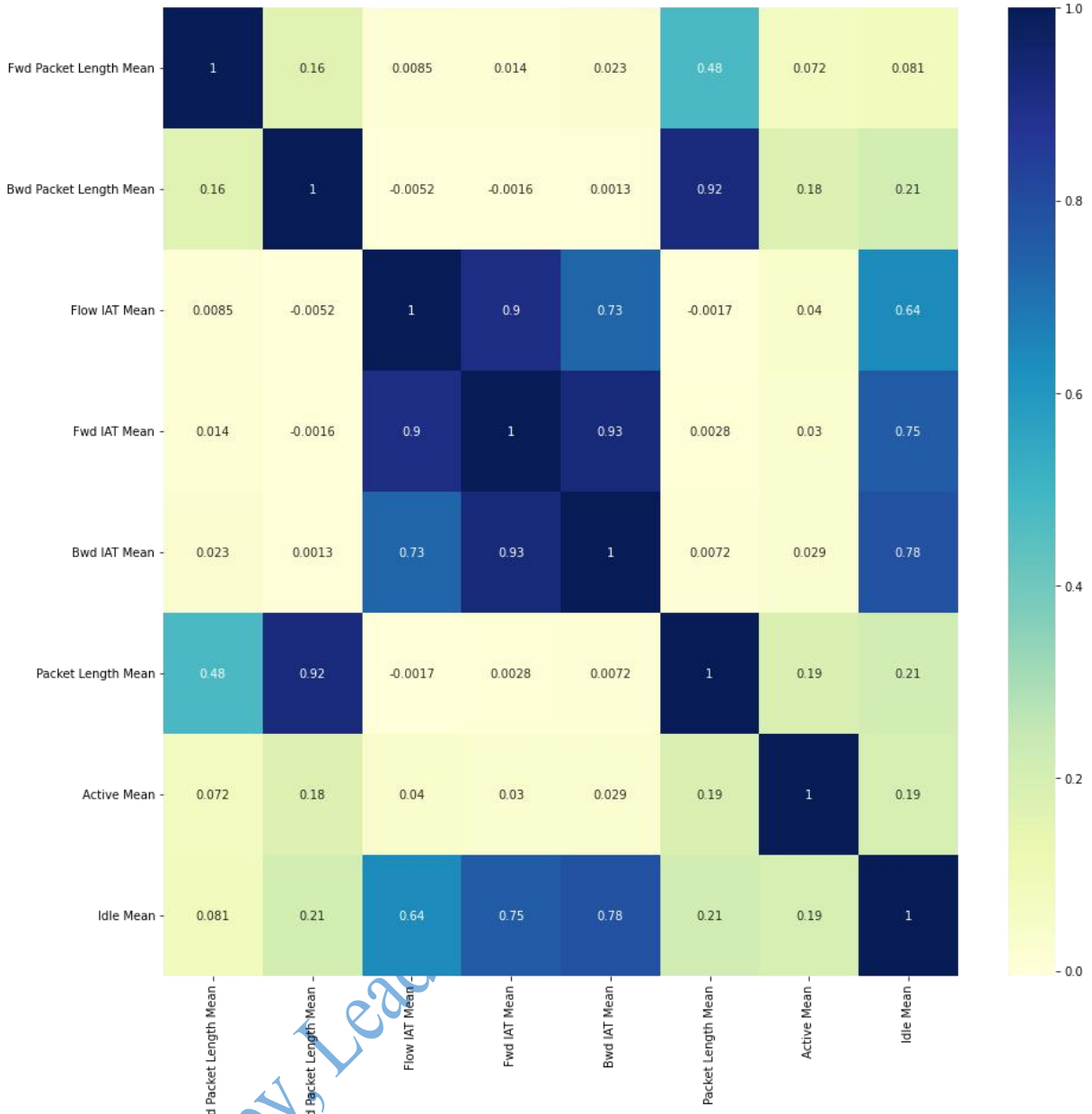
In [30]:
```
# Remove columns that have corr > 0.7 in heatmap
cols_to_drop=['Bwd  PSH Flags', 'Fwd URG Flags', 'Bwd URG Flags']
drop_cols(cols_to_drop  )
```

In [31]:
```
# Check correlation in aggregated max columns
cols = [x for x in df.columns if 'Max' in x]

plot_corr(cols)
```



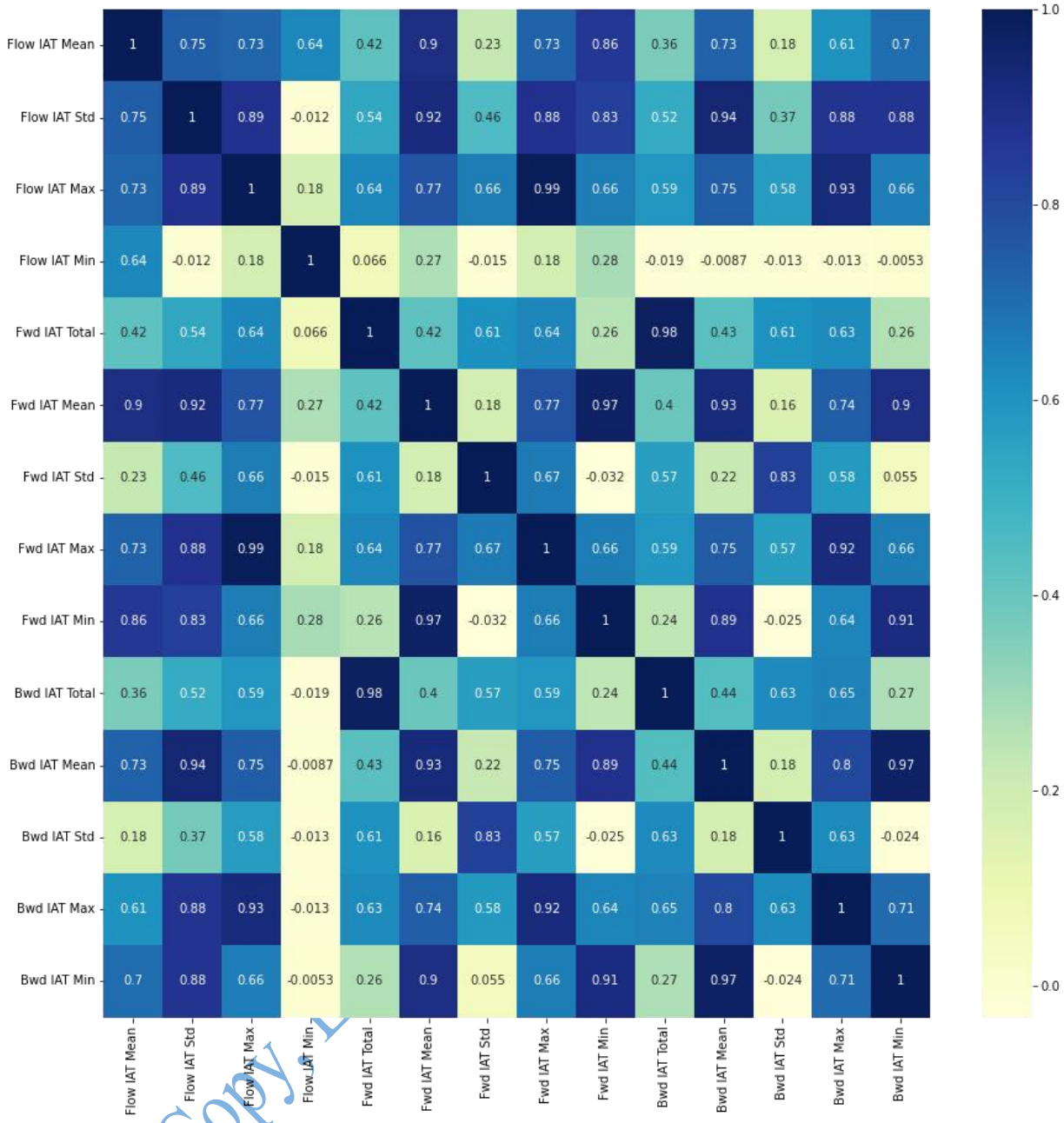
In [32]:
```
# Remove columns that have corr > 0.7 in heatmap
cols_to_drop=['Bwd Packet Length Max','Fwd IAT Max','Bwd IAT Max','Idle Max']
drop_cols(cols_to_drop)
```

In [33]:
```
# Check correlation in aggregated mean columns
cols = [x for x in df.columns if 'Mean' in x]

plot_corr(cols)
```



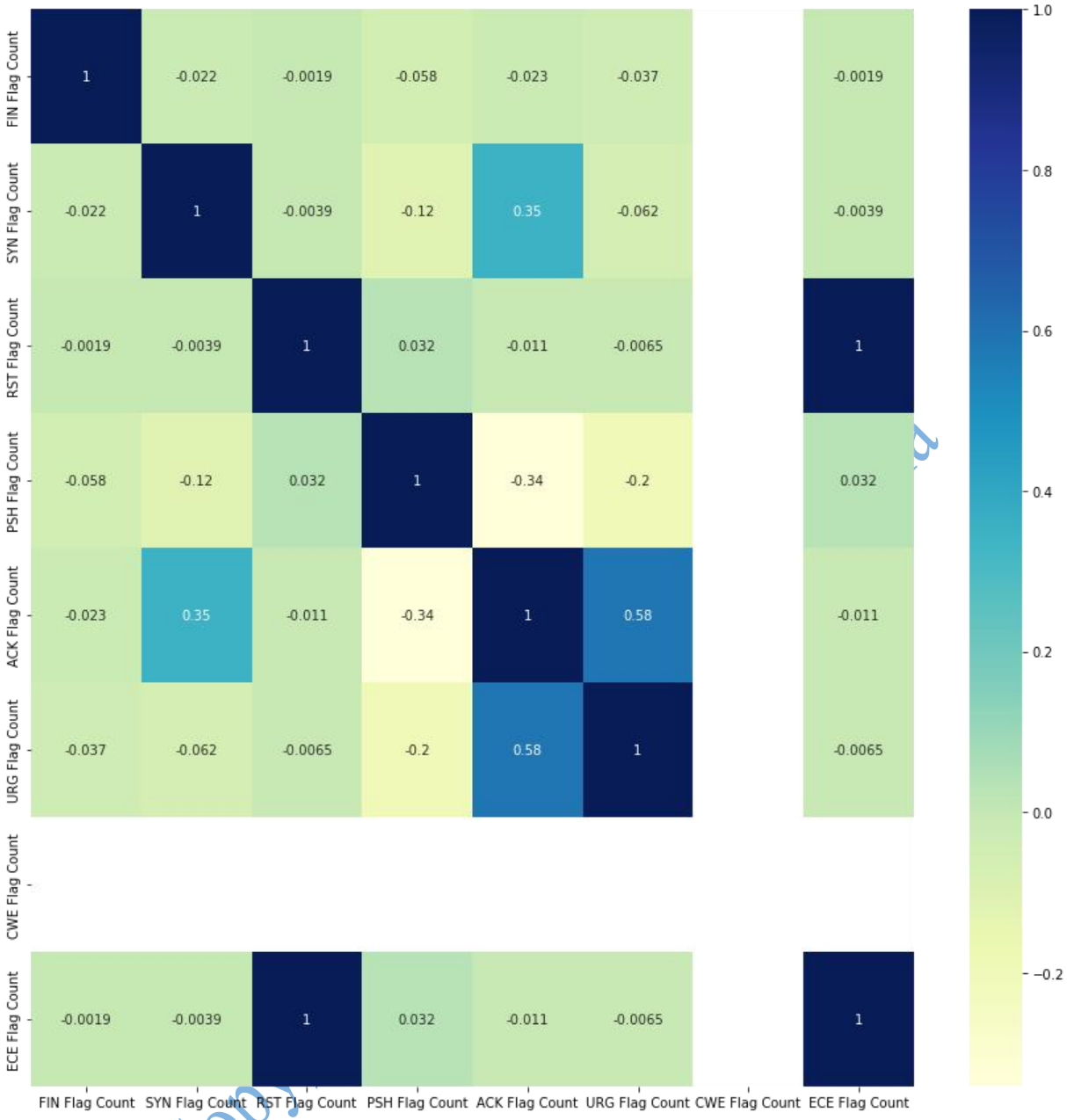
In [34]:
```
# Remove columns that have corr > 0.7 in heatmap
cols_to_drop=['Bwd Packet Length Mean','Fwd IAT Mean','Bwd IAT Mean']
drop_cols(cols_to_drop)
```

In [35]:
```
# Check correlation in columns that contain IAT
cols = [x for x in df.columns if 'IAT' in x]

plot_corr(cols)
```



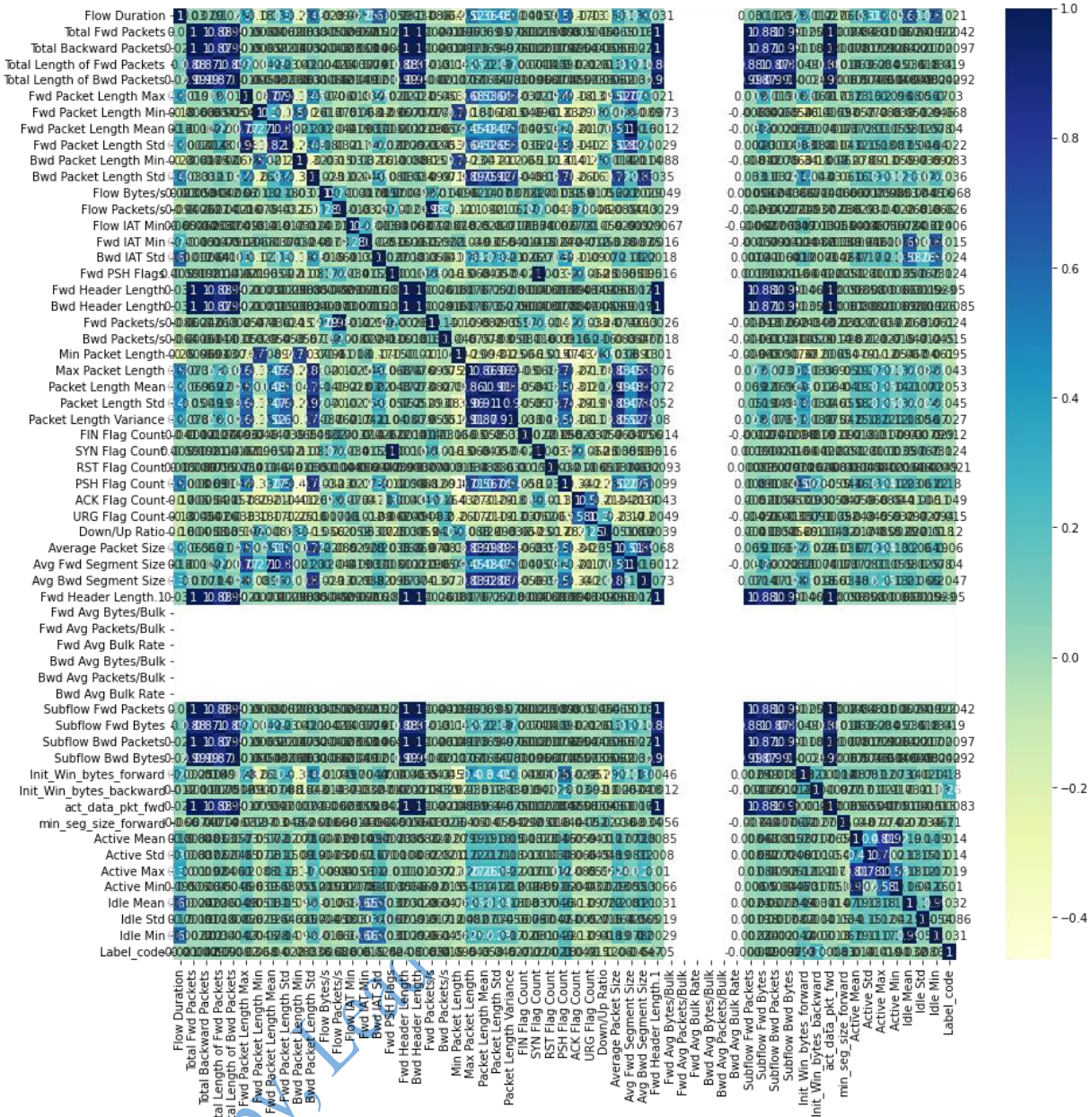
In [36]:
```
# Remove columns that have corr > 0.7 in heatmap
cols_to_drop=['Fwd IAT Std','Bwd IAT Min','Flow IAT Mean','Flow IAT Std','Flow IAT M
drop_cols(cols_to_drop)
```

In [37]:
```
# Check correlation in aggregated count columns
cols = [x for x in df.columns if 'Count' in x]

plot_corr(cols)
```



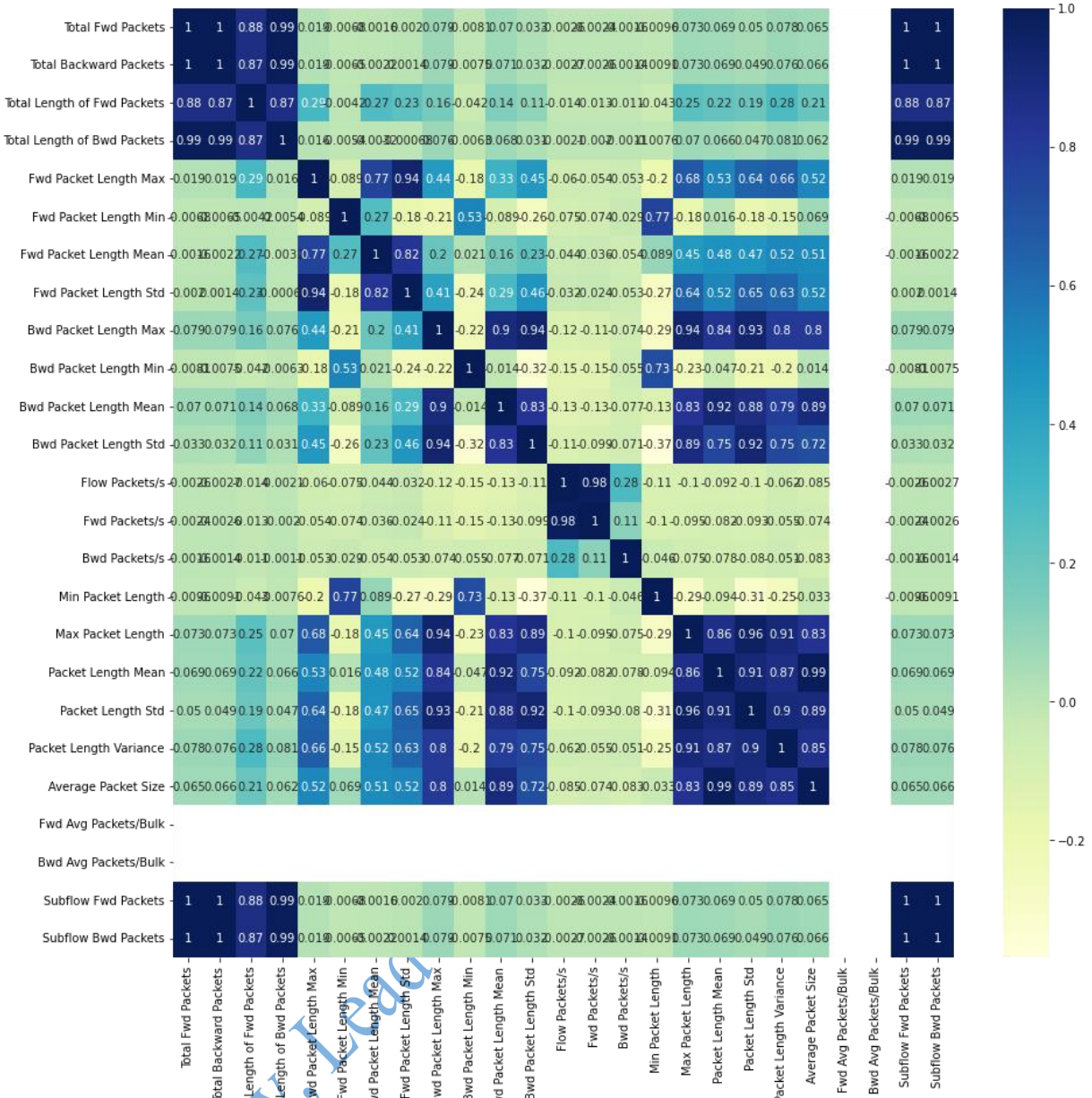
In [38]:
```
# Remove columns that have corr > 0.7 in heatmap
cols_to_drop=['CWE  Flag  Count','ECE  Flag  Count']
drop_cols(cols_to_drop)
```

In [39]:
```
plot_corr(features)
```



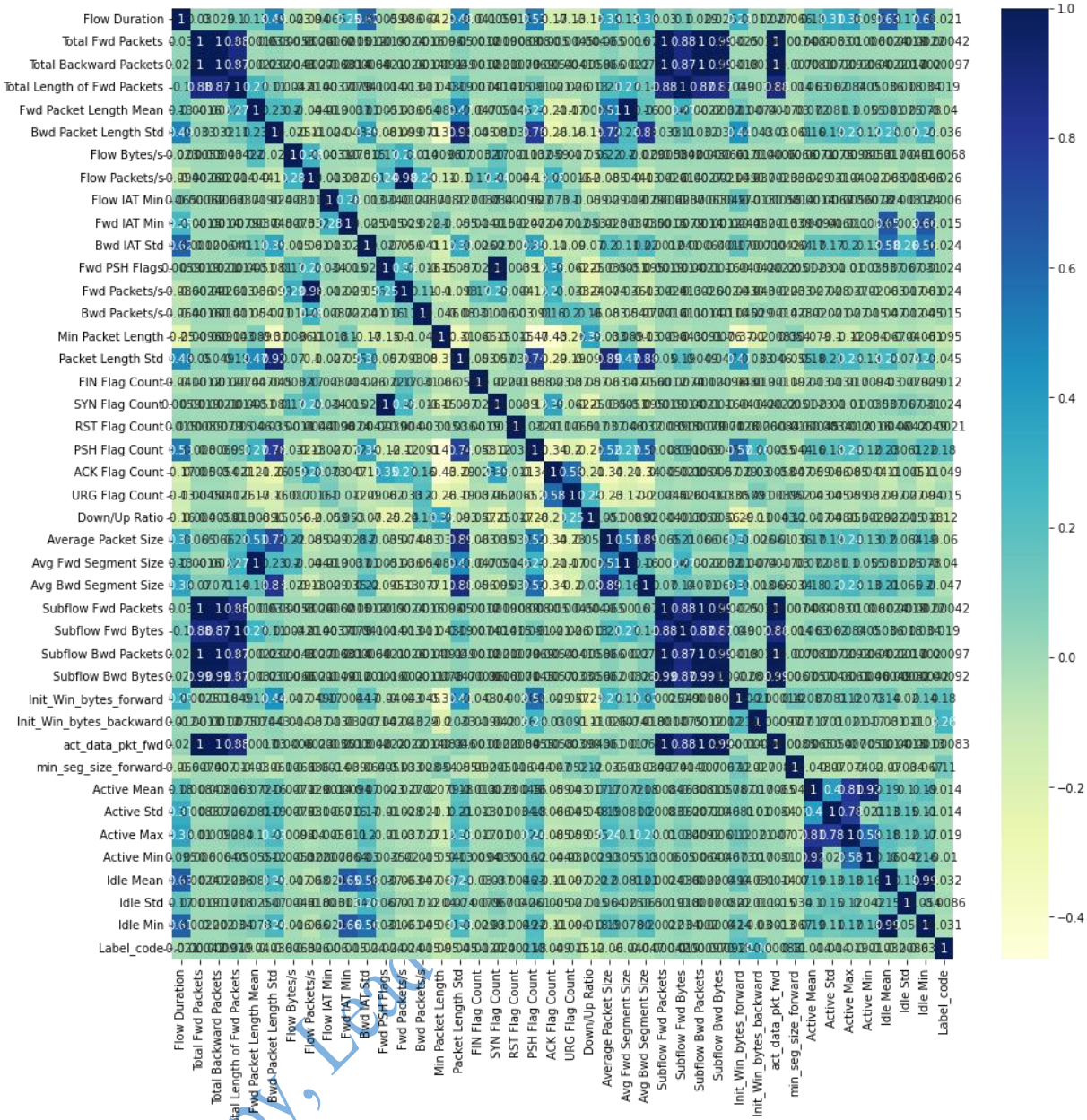
In [40]:
```
cols_to_drop=['Fwd  Avg Bytes/Bulk','Fwd  Avg Packets/Bulk','Fwd  Avg Bulk Rate', 'Bwd
              'Bwd Avg Packets/Bulk','Bwd  Avg Bulk Rate']
drop_cols(cols_to_drop)
```

In [41]:
```
# Check correlation in aggregated count columns
cols = [x for x in df.columns if 'Packet' in x]

plot_corr(cols)
```



In [42]:
```
cols_to_drop=['Total  Length of Bwd Packets','Fwd Header Length','Bwd Header Length',
'Fwd Packet Length Min','Fwd Packet Length Std','Packet Length Variance','Max Packet
 'Packet Length Mean','Fwd Packet Length Max','Bwd Packet Length Min']
drop_cols(cols_to_drop)
```

In [43]:
```
plot_corr(features)
```

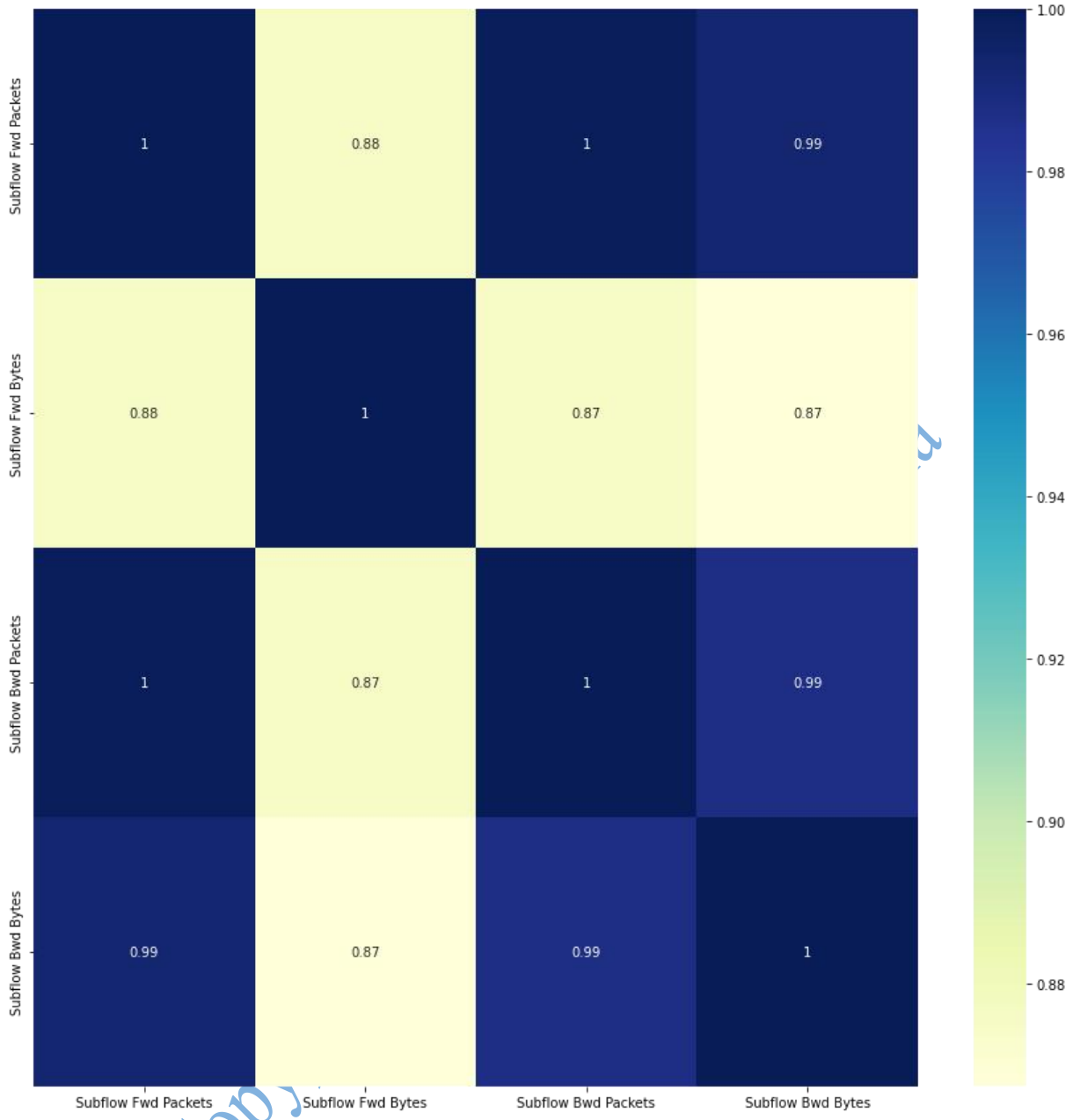


In [44]:

```
cols_to_drop = ['Total Backward Packets','Total Length of Fwd Packets']
drop_cols(cols_to_drop)
```

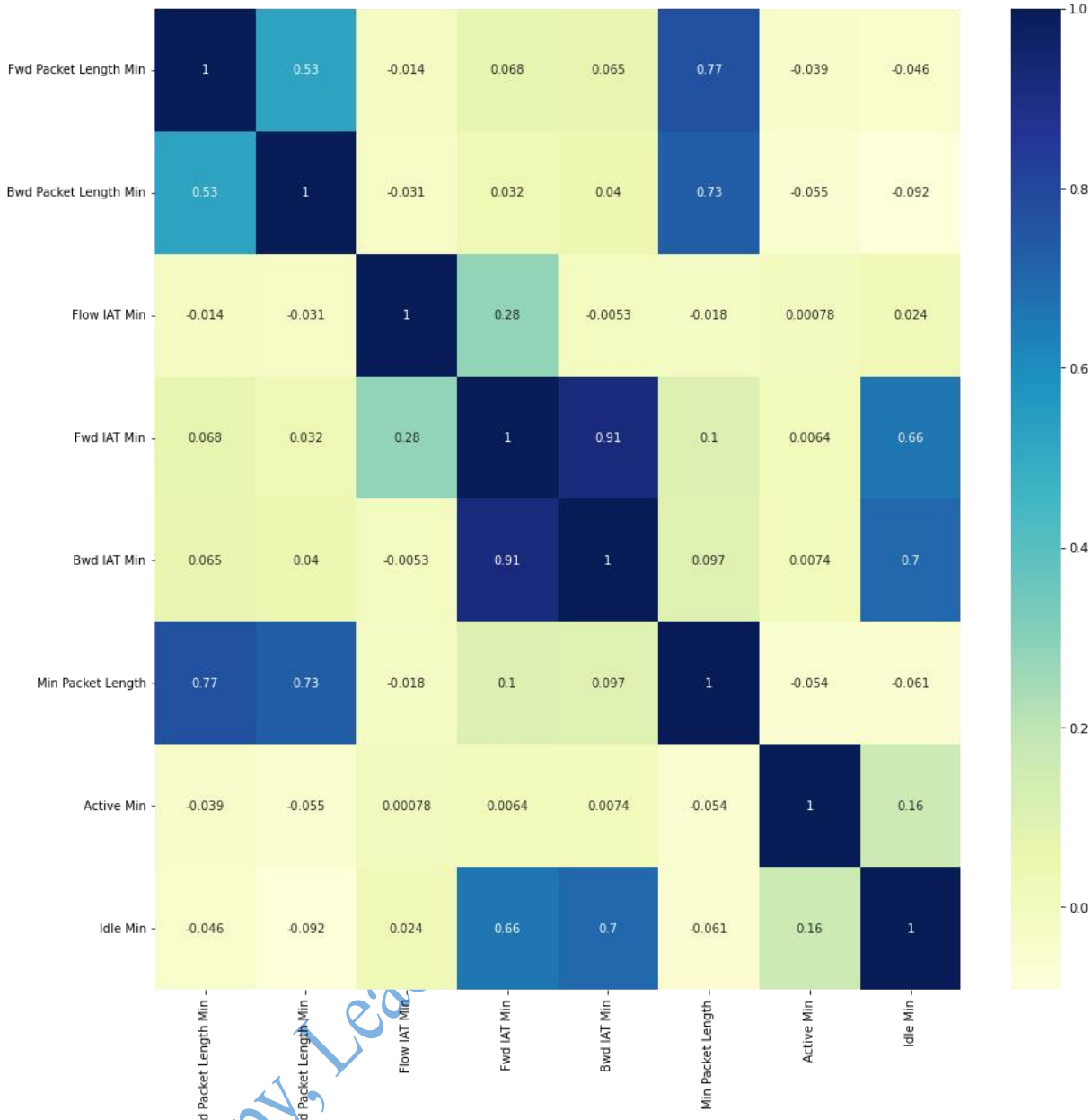
In [45]:

```
# Check correlation in aggregated count columns
cols = [x for x in df.columns if 'Subflow' in x]

plot_corr(cols)
```



In [46]:
```
cols_to_drop=['Subflow  Fwd Packets','Subflow  Fwd Bytes','Subflow  Bwd Packets']
drop_cols(cols_to_drop)
```

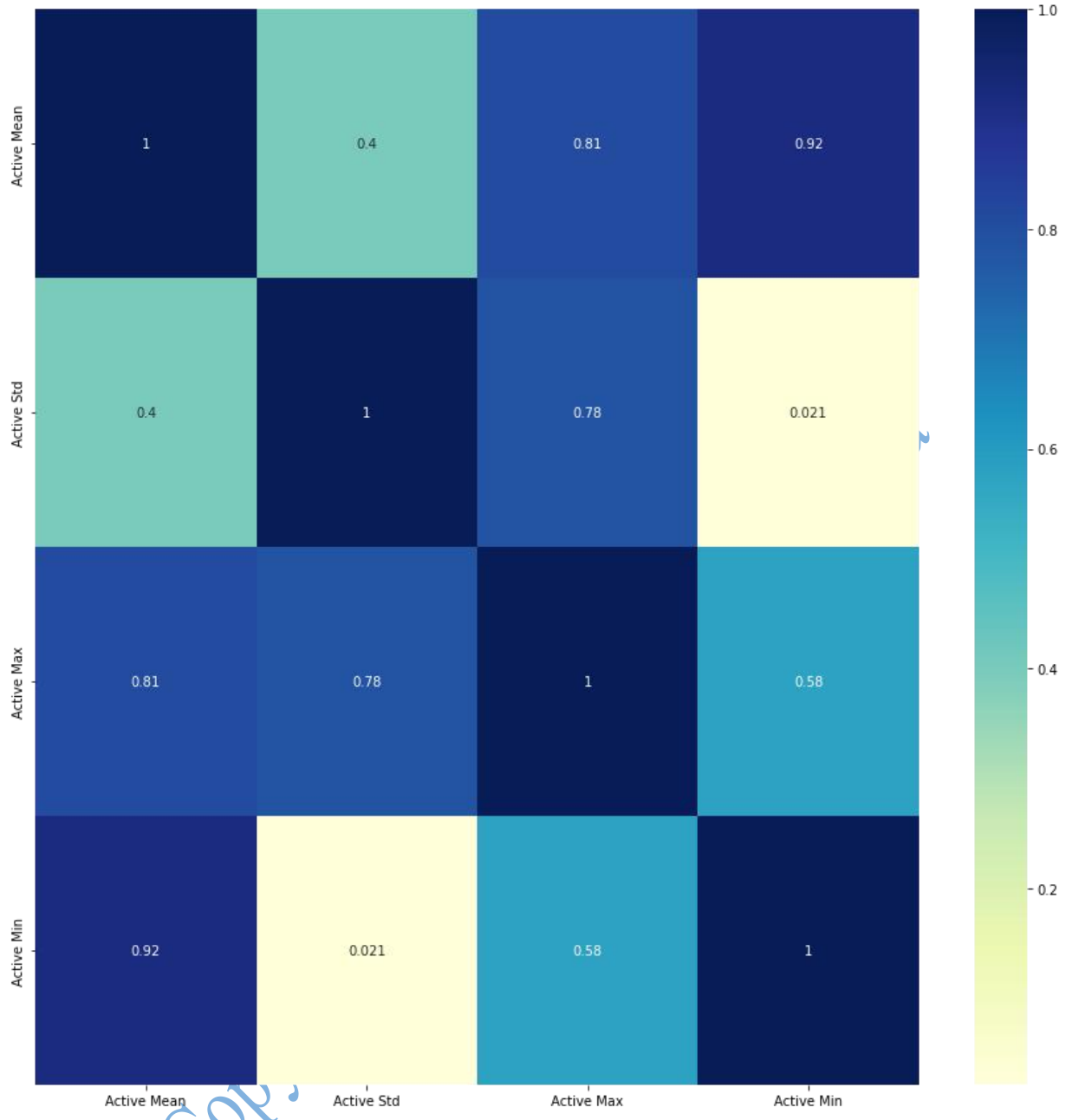
In [47]:
```
# Check correlation in aggregated count columns
cols = [x for x in df.columns if 'Min' in x]

plot_corr(cols)
```



In [48]:
```
cols_to_drop= ['Min Packet Length','Bwd IAT Min','Idle Min']
drop_cols(cols_to_drop)
```

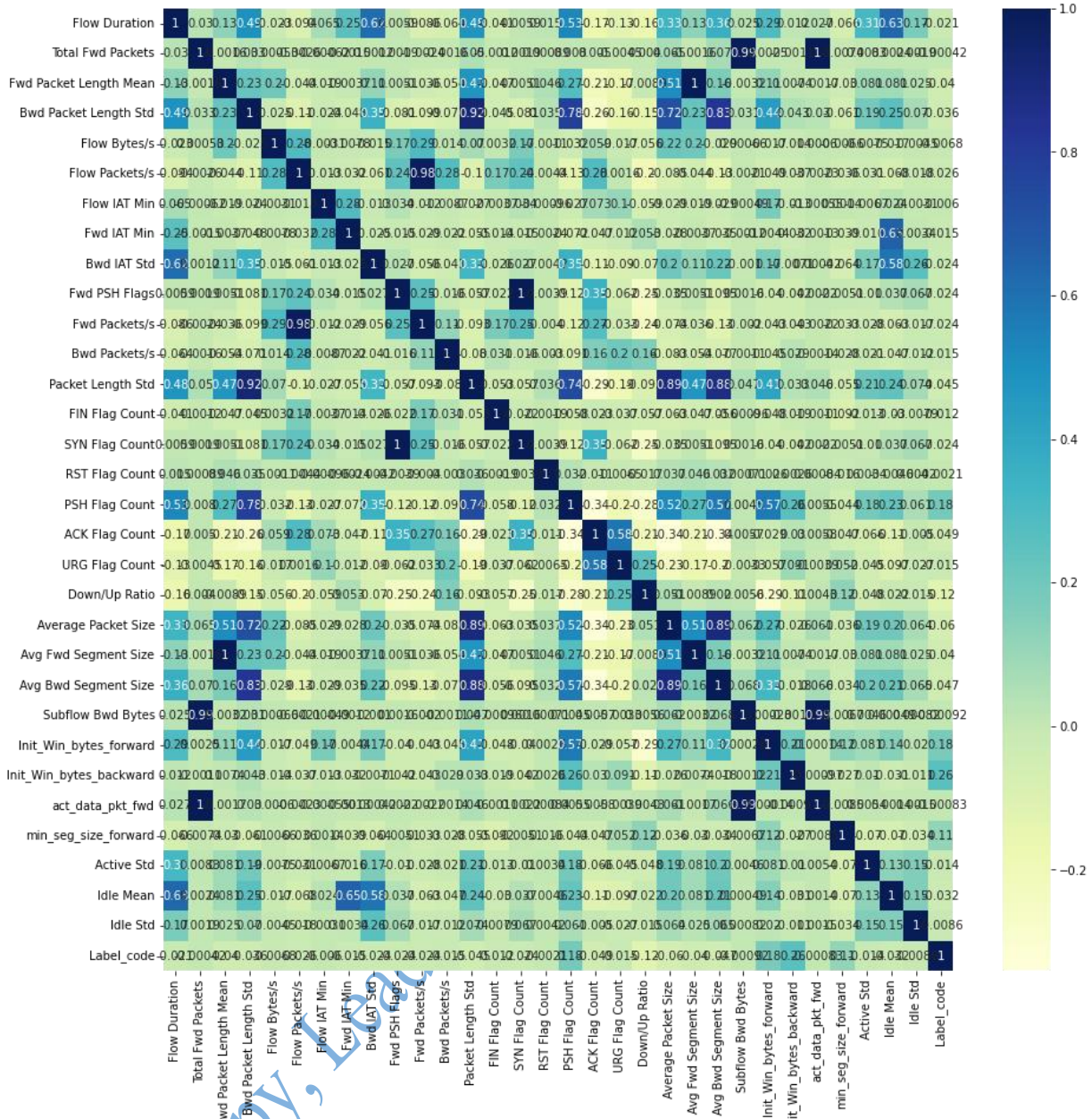
```
Unable to remove Bwd IAT Min
```

In [49]:
```
# Check correlation in aggregated count columns
cols = [x for x in df.columns if 'Active' in x]

plot_corr(cols)
```



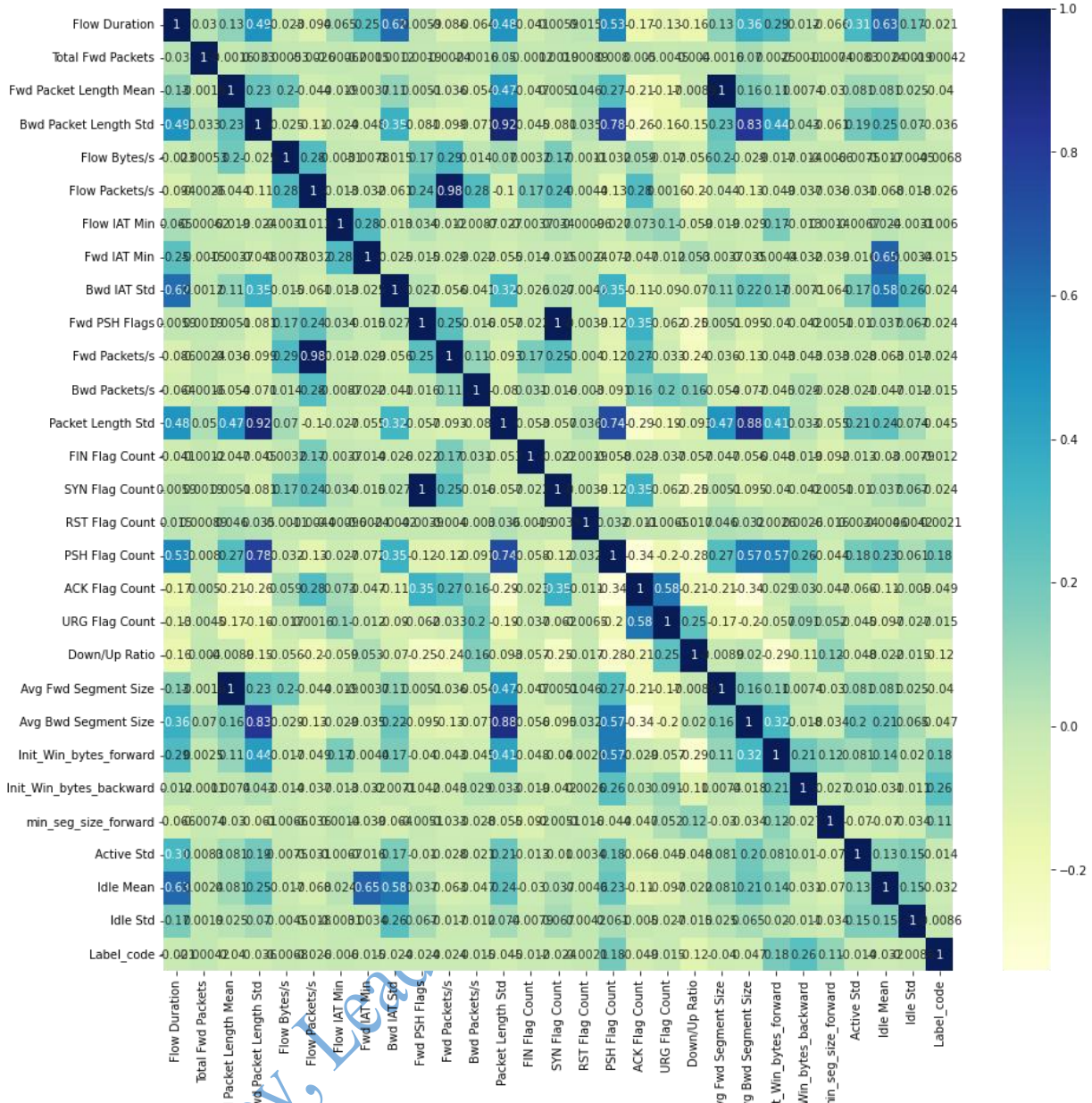
In [50]:
```
cols_to_drop= ['Active Max','Active Min','Active Mean']
drop_cols(cols_to_drop)
```

In [51]:
```
plot_corr(features)
```



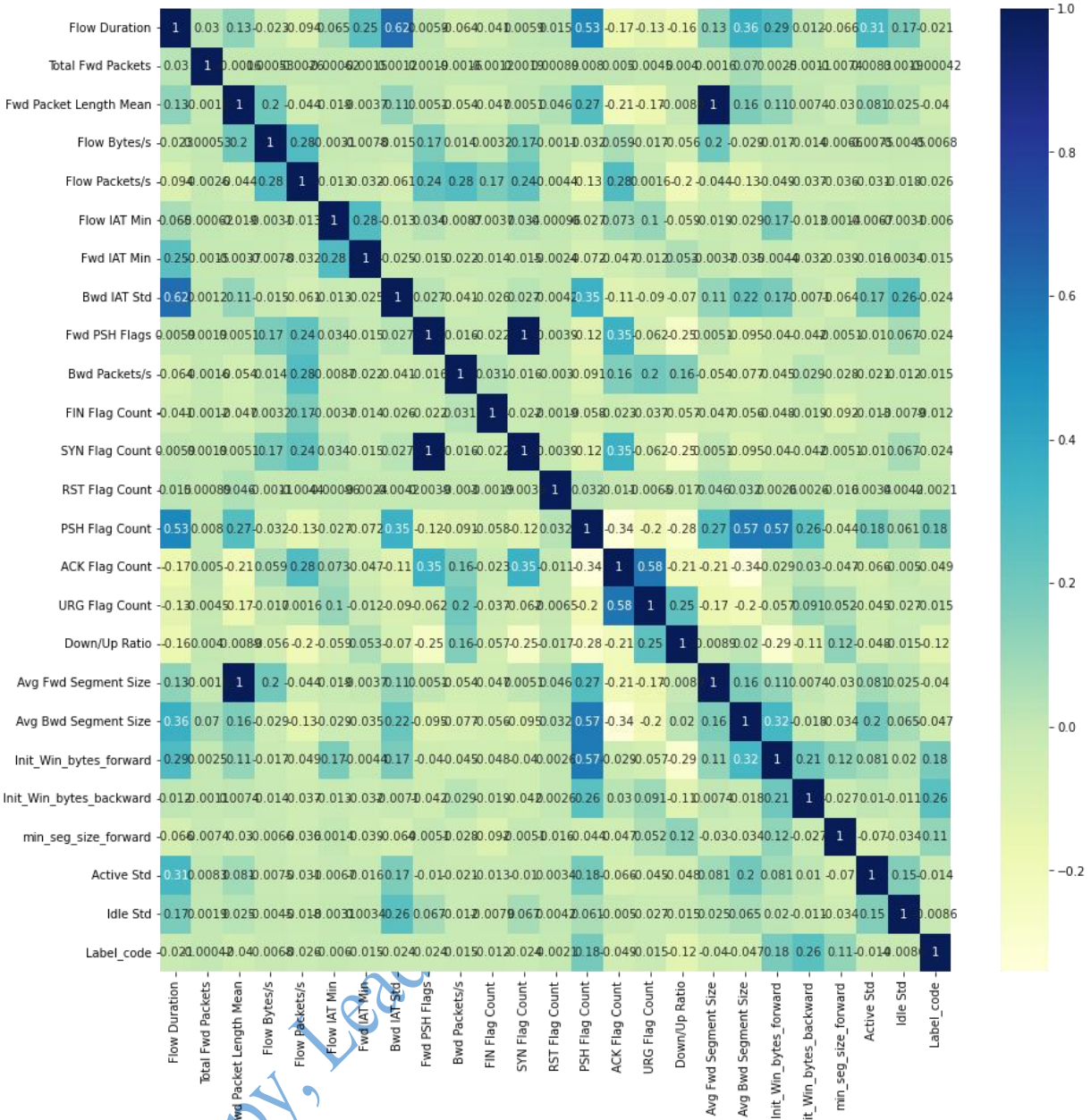
In [52]:
```
cols_to_drop= ['Subflow Bwd Bytes','act_data_pkt_fwd','Average  Packet Size']
drop_cols(cols_to_drop)
```

In [53]:
```
plot_corr(features)
```



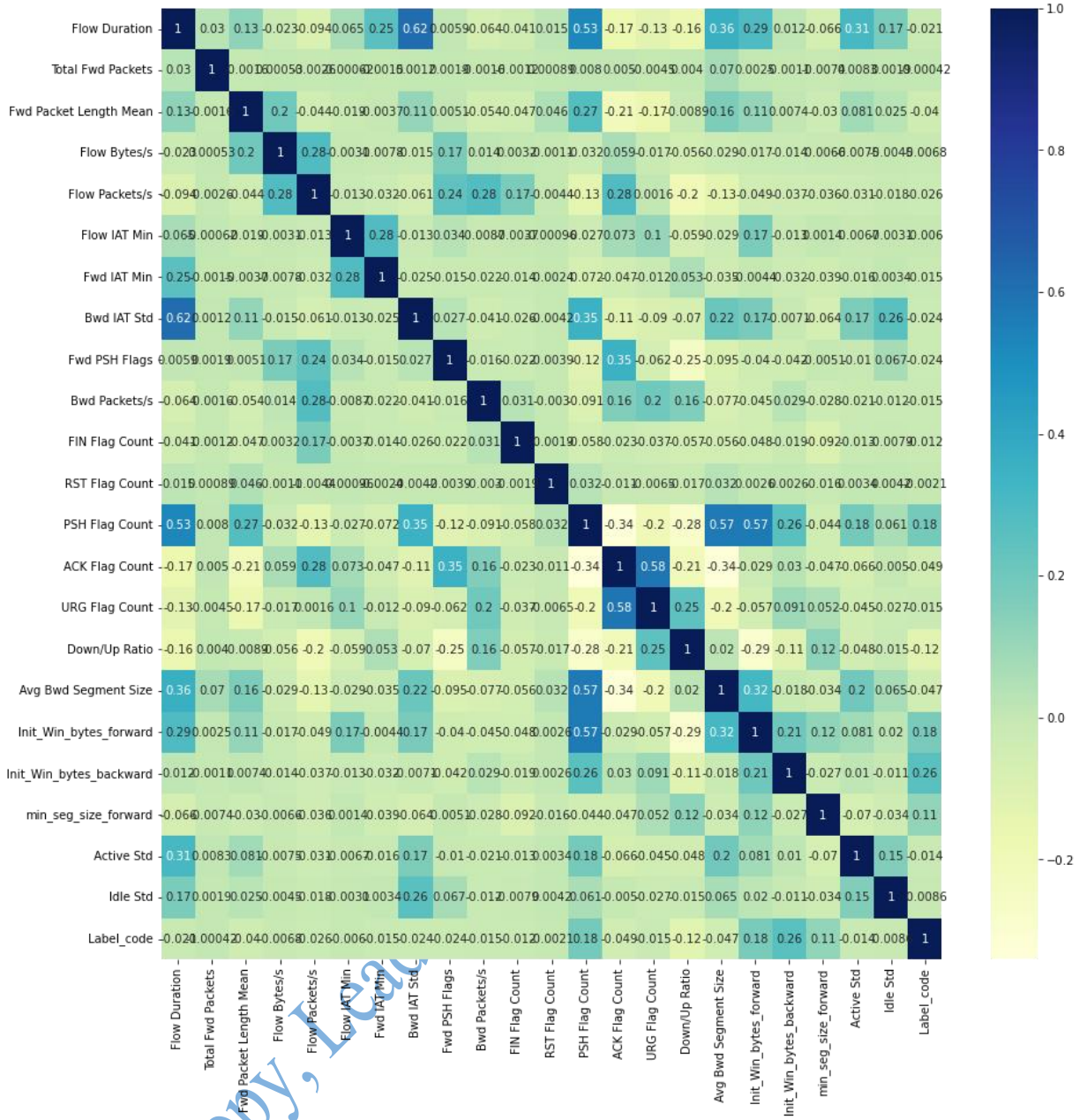
In [54]:
```
cols_to_drop = ['Packet Length Std','Idle Mean','Fwd Packets/s','Bwd Packet Length S
drop_cols(cols_to_drop)
```

In [55]:
```
plot_corr(features)
```



In [56]:
```
cols_to_drop =['SYN Flag Count','Avg Fwd Segment Size']
drop_cols(cols_to_drop)
```

In [57]:
```
plot_corr(features)
```



cols_to_drop = ['Bwd Packet Length Mean','Fwd IAT Mean','Bwd IAT Mean','Bwd Packet Length Max','Fwd IAT Max','Bwd IAT Max','Idle Max',\ 'Bwd Packet Length Std','Fwd IAT Std','Min Packet Length','Bwd IAT Min','Idle Min','Fwd Avg Bytes/Bulk','Fwd Avg Packets/Bulk',\ 'Fwd Avg Bulk Rate', 'Bwd Avg Bytes/Bulk', 'Bwd Avg Packets/Bulk','Bwd Avg Bulk Rate','ECE Flag Count',\ 'Total Length of Bwd Packets','Fwd Header Length','Bwd Header Length','Fwd Header Length.1', \ 'Fwd Packet Length Min','Fwd Packet Length Std','Packet Length Variance','Max Packet Length',\ 'Packet Length Mean','Fwd Packet Length Max','Bwd Packet Length Min','Flow IAT Mean','Flow IAT Std','Flow IAT Max',\ 'Bwd IAT Total','Bwd PSH Flags', 'Fwd URG Flags', 'Bwd URG Flags','CWE Flag Count','Fwd Avg Bytes/Bulk','Idle Mean',\ 'Total Backward Packets','Total Length of Fwd Packets','Avg Fwd Segment Size','act_data_pkt_fwd','PSH Flag Count',\ 'Packet Length

Std','Average Packet Size', 'Active Max','Active Min','Active Mean','SYN Flag Count',\ 'Subflow Fwd Packets','Subflow Fwd Bytes','Subflow Bwd Packets','Subflow Bwd Bytes','Fwd IAT Total','Fwd Packets/s']

In [58]:
```
features
```

Out[58]:
```
['Flow Duration',
 'Total Fwd Packets',
 'Fwd Packet Length Mean',
'Flow Bytes/s',
'Flow Packets/s',
'Flow IAT Min',
'Fwd IAT Min',
'Bwd IAT Std',
'Fwd PSH Flags',
'Bwd Packets/s',
'FIN Flag Count',
'RST Flag Count',
'PSH Flag Count',
'ACK Flag Count',
'URG Flag Count',
'Down/Up Ratio',
'Avg Bwd Segment Size',
'Init_Win_bytes_forward',
'Init_Win_bytes_backward',
'min_seg_size_forward',
'Active Std',
'Idle Std',
'Label',
'Label_code']
```

Get data with best features

Observation : columns was reduced from 79 to 24

In [59]:
```
model_df = df[features]
```

In [60]:
```
model_df.head(3)
```

Out[60]:

	Flow Duration	Total Fwd Packets	Fwd Packet Length Mean	Flow Bytes/s	Flow Packets/s	Flow IAT Min	Fwd IAT Min	Bwd IAT Std	Fwd PSH Flags	Bwd Packets/s
0	113095465	48	201.416667	174.012282	0.636630	3	3	7084368.263	1	0.212210
1	113473706	68	167.117647	212.225377	0.951762	2	2	5922355.273	1	0.352505
2	119945515	150	0.000000	0.000000	1.250568	0	0	0.000	0	0.000000

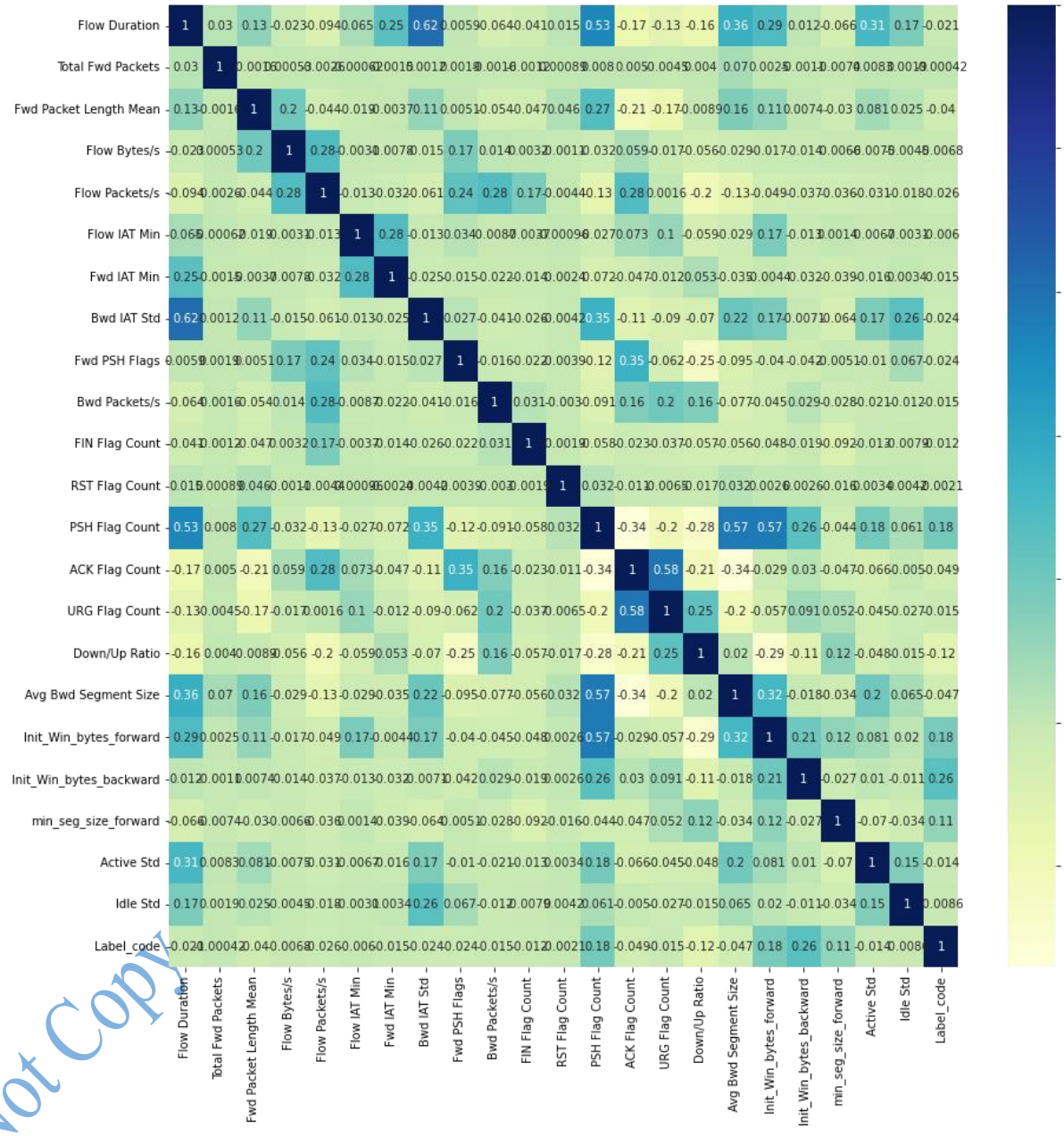
In [61]:
```
model_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 170366 entries, 0 to 170365
Data columns (total 24 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   Flow Duration           170366 non-null  int64
 1   Total Fwd Packets       170366 non-null  int64
 2   Fwd Packet Length Mean  170366 non-null  float64
 3   Flow Bytes/s            170366 non-null  float64
 4   Flow Packets/s          170366 non-null  float64
 5   Flow IAT Min            170366 non-null  int64
 6   Fwd IAT Min             170366 non-null  int64
 7   Bwd IAT Std             170366 non-null  float64
 8   Fwd PSH Flags           170366 non-null  int64
 9   Bwd Packets/s           170366 non-null  float64
 10  FIN Flag Count          170366 non-null  int64
```

```
11  RST Flag Count           170366 non-null  int64
```

```
12  PSH Flag Count            170366 non-null  int64
13  ACK Flag Count            170366 non-null  int64
14  URG Flag Count            170366 non-null  int64
15  Down/Up Ratio             170366 non-null  int64
16  Avg Bwd Segment Size      170366 non-null  float64
17  Init_Win_bytes_forward    170366 non-null  int64
18  Init_Win_bytes_backward   170366 non-null  int64
19  min_seg_size_forward      170366 non-null  int64
20  Active Std                170366 non-null  float64
21  Idle Std                  170366 non-null  float64
22  Label                     170366 non-null  object
 23  Label_code                170366 non-null
float64 dtypes: float64(9), int64(14),
object(1)
memory usage: 31.2+ MB
```

```
In [62]:  plot_corr(features, model_df)
```


Flow Duration
Total Fwd Packets
Fwd Packet Length Mean
Flow Bytes/s
Flow Packets/s
Flow IAT Min
Fwd IAT Min
Bwd IAT Std
Fwd PSH Flags
Bwd Packets/s
FIN Flag Count
RST Flag Count
PSH Flag Count
ACK Flag Count
URG Flag Count
Down/Up Ratio
Avg Bwd Segment Size
Init_Win_bytes_forward
Init_Win_bytes_backward
min_seg_size_forward
Active Std
Idle Std
Label_code

Appendix D: Export Model Data

```
In [63]: model_df.to_csv('model_data.csv', index=False)
```

See target distribution

```
In [64]: model_df['Label_code'].value_counts()
```

```
Out[64]: 0.0    168186
         1.0      2180
         Name: Label_code, dtype: int64
```

Get Features and Target

```
In [65]: features =
         model_df.columns.t
         o_list()
         features.remove('L
         abel')
         features.remove('L
         abel_code')
```

```
In [66]: X = model_df[features]
         Y = model_df['Label_code']
         del model_df
```

```
In [67]: from sklearn.metrics import accuracy_score
         from sklearn.metrics import
         classification_report from
         sklearn.model_selection import
         train_test_split from
         sklearn.ensemble import
         RandomForestClassifier
```

Split dataset into train and test set

```
In [68]: #set stratify=Y because the target values are not evenly distributed
         X_train, X_test, Y_train, Y_test =
         train_test_split(X,Y,test_size=0.2,random_state=0
```

In [69]:
```
X_train.head(5)
```

Flow	Duration	Total Fwd Packets	Fwd Packet Length Mean	Flow Bytes/s	Flow Packets/s	Flow IAT Min	Fwd IAT Min	Bwd IAT Std	Fwd PSH Flags
119479	185	2	41.000	1.816216e+06	21621.621620	3	3	0.000	0
116026	116072159	32	27.125	3.043538e+02	0.585842	1	3	4805501.762	0
64567	146479	2	0.000	0.000000e+00	20.480752	54	146479	0.000	0
19529	160	2	45.000	1.325000e+06	25000.000000	1	1	0.000	0
96325	24659	2	6.000	7.299566e+02	121.659435	3	3	0.000	0

[redacted]

Normalization

Scaling to reduce feature data scale and handle values too large for float32

In [70]:
```
from sklearn.preprocessing
import StandardScaler scaler
= StandardScaler()
# fit the scale
scaler_fit = scaler.fit(X_train)
# transformation of training data
scal_xtrain = scaler_fit.transform(X_train)
# transformation of testing data
scal_xtest = scaler_fit.transform(X_test)
```

In [71]:
```
scal_xtrain
```

Out[71]:

```
array([[-0.39145132, -0.01149707, -0.07595572, ...,  1.01002916,
        -0.12419664, -0.07576214],
       [ 3.23492064,  0.01394101, -0.2225462 , ..., -0.87849447,
         1.23510211, -0.07252062],
       [-0.38688074, -0.01149707, -0.50912398, ...,  1.01002916,
        -0.12419664, -0.07576214],
       ...,
       [ 0.55708438, -0.00471358,  2.59806595, ..., -0.87849447,
         0.24833414, -0.00328719],
       [-0.39145008, -0.01234501, -0.50912398, ...,  1.01002916,
        -0.12419664, -0.07576214],
       [-0.39145254, -0.01149707, -0.18160651, ...,  1.01002916,
        -0.12419664, -0.07576214]])
```

Appendix E: Model Fitting and Testing Predictions

Our focus target value (web attack,1) is about 1/39 of all target values.

For this project, the aim is to predict correctly if a Network flow is a Webattack

beacuse ... So the accuracy metric for model selection will be recall. High recall means

low false negatives

Defining recall with respect to this project: Recall gives the fraction of network

flows correcly identified as web attacks out of all web attacks.

Mathematically,

Recall = True Positive / (True Positive

+ False Negative) Ensemble Modelling using Random Forest

Observation: Recall for web attack (1) is 0.98

```
In [72]: #Fitting Random forest classification to training set
         RFclassifier = RandomForestClassifier(  n_estimators = 100,criterion=
         'entropy',class
         RFclassifier.fit(scal_xtrain,  Y_train
```

```
Out[72]:RandomForestClassifier(class_weight='balanced_subsample',
criterion='entropy')
```

```
In [73]: # Predicting Test set results
         Y_pred = RFclassifier.predict(scal_xtest)

         # Evaluate model
         print(f'Random Forest Accuracy score: {round(accuracy_score(Y_test,
         Y_pred),4)}')

         print(classification_report(Y_test,  Y_pred))
```

```
Random Forest Accuracy score: 0.9997

              precision    recall  f1-score   support

         1.0       1.00      0.98      0.99       436

    accuracy                           1.00     34074
   macro avg       1.00      0.99      0.99     34074
weighted avg       1.00      1.00      1.00     34074
```

"The dataset column names contains a lot of messy data, model fitting with raw data is impossible due to value errors."

]

},

{

"cell_type": "code",

"execution_count": 1,

"id": "539b0049",

"metadata": {},

"outputs": [],

"source": [

"import pandas as pd\n",

"import numpy as np \n",

"import warnings\n",

"warnings.filterwarnings('ignore')\n",

```
  "from datetime import datetime\n",
  "from sklearn.metrics import accuracy_score\n",
  "from sklearn.metrics import classification_report\n",
  "from sklearn.model_selection import train_test_split\n",
  "from sklearn.ensemble import RandomForestClassifier"
 ]
},
{
 "cell_type": "code",
 "execution_count": 2,
 "id": "4c3a5f2b",
 "metadata": {},
 "outputs": [],
 "source": [
  "initial_df = pd.read_csv('MachineLearningCVE/Thursday-WorkingHours-Morning-
WebAttacks.pcap_ISCX.csv')"
 ]
},
{
```

"cell_type": "code",

"execution_count": 3,

"id": "b1f1345a",

"metadata": {

"scrolled": true

},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 170366 entries, 0 to 170365\n",

Biodata

A. Personal Information

Surname:	Oluwaseye Abayomi ADEYEMI
Date of Birth:	17th March, 1985
Sex:	Male
Place of Birth:	Ikere-Ekiti
State of Origin:	Ekiti – State
Local Govt.:	Ikere Ekiti Local Govt. Area
Nationality:	Nigerian
Postal Address:	Behind Endurance furniture shop Kajola area, Ona Obe street Ikere Ekiti
Phone Number:	07062284812, 07014565433
Email:	sheyee4u2@gmail.com,
Residential Address:	Behind Endurance furniture shop Kajola area, Ona Obe street Ikere Ekiti
Permanent Home	
Town Address:	Osolo Compound Oke Ikere, Ikere Ekiti
Next of Kin:	Adeyemi Blessing Grace Oluwatimilehin
Address of Next of Kin:	Behind Endurance furniture shop Kajola area, Ona Obe street, Ikere Ekiti

B. Educational Background with Dates

Lead City University Ibadan, Oyo State	2019 – till date
Lead City University Ibadan, Oyo State	2014 - 2017
School of Health Information Management,	
University of Ilorin Teaching Hospital, Ilorin Kwara State	2010 – 2012
College of Health Technology Ilesa, Osun State	2007 – 2010
Aipate Baptist Church Grammar School Iwo, Osun State	2001 – 2006
Methodist Primary School Araromi Iwo, Osun State	1996 – 2001

C. Academic Qualifications with and Certificate Obtained with Dates

M.Sc. in Computer Science	2019 – till date
B.Sc. in Health Information Management	2014 – 2017
Higher National Diploma in Health Information Management	2010 – 2012
Health Records Technician in Health information Management	2007 – 2010
West African School Certificate	2001 – 2006
First School Leaving Certificate	1996 – 2001

D. Working Experience with Dates

Fabotas College of Health Sciences and Technology Ado Ekiti, Ekiti state 2013 – till date

Health Information Management Lecturer

WECARE College of Health Sciences and Technology Iperu Remo, Ogun State Oct. 2020 – Dec. 2020

Health Information Management Lecturer

APIN INITIATIVES(NGO) 2020- 2021

Health Informatics Clerk/ Data Entry Assistant

BOUESTI 2022 – till date

Graduate Assistant

E. Conference/Seminars/Workshops Attended with Dates

National Health Information Management System,

Chida International Hotel, Abuja 2021

Professional Instructors & Academicians, **Abuja** 2020

Annual National Scientific Conference and AGM,

Sir Ahmadu Bello Hall New Secretariat Complex Dutse Jigawa State 2018

Seminar & Scientific Workshop by Health Information Managers Association,

Conference Hall, Shiroro Hotel, Minna, Niger State 2011

Areas of Research Interests and Activities

Electronic Health Records, Information System, Machine Learning, Health information Management and making new friends.

Membership of Academic Professional Bodies

Associate Member of the Guild of Professional Instructor and Academicians of Nigeria 2020

F. References

Dr. (Associate Professor) K. S. Osundina

Head of Department of Health Information Management,

Adeleke University Ede,

Osun State.

Tel: 08033852660

Dr. A.A. Waheed

Computer Science Department,

Lead City University, Ibadan,

Oyo State.

Tel: 07031199441

Dr. S.M. Omole

Head of Department of Health Information Management,

College of Health Technology, Ilesa,

Osun State.

Tel: 08035896256

Signature **Date**

The University Compliance Certification

This is to certify that this thesis by Oluwaseye Abayomi ADEYEMI with Matriculation Number LCU/PG/001144 in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan is in full compliance with the approved University's Format and Style.

Signature	**Date**