Detection of DDOS Attacks in A Smart Building Using ML

By

Yumna Nasir

00000327052

Supervisor

Assistant Professor Dr Yawar Abbas Bangash

A thesis submitted in the department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan for the partial fulfilment of the requirement for degree of MS in Software Engineering

June,2023

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Mrs. Yumna Nasir**, Registration No. **00000327052**, of **Military College of Signals** has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations/MS Policy, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members and local evaluators of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor Assistant Professor Dr Yawar

Date: 1/8/23

Signature (HOD): _____

Brig
Head of Dept of CSE
Mil College of Sigs (NUST)

Date: 21/8/23

Signature (Dean/Principal) _____

Date: 21/8/23

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

ABSTRACT

Smart buildings, enabled by the widespread use of Internet of Things (IoT) devices, are becoming increasingly prevalent. However, this rise in IoT adoption also brings new security challenges, with smart buildings more vulnerable to cyber-attacks, including distributed denial-of-service (DDoS) attacks. DDoS attacks can cause significant damage to the building's network infrastructure, leading to financial losses and downtime.

This thesis proposes a machine learning (ML) based approach to detect DDoS attacks in smart buildings. The proposed solution employs various ML algorithms, including SVM, decision trees, Neural Network using TensorFlow and linear regression. These models are trained to analyse network traffic data collected from smart building devices and detect and classify network traffic patterns that indicate DDoS attacks.

To train the ML models, network traffic data collected from smart buildings is pre-processed to extract relevant features. The performance of the models is evaluated based on accuracy, precision, and recall metrics. The results show that the proposed ML-based approach outperforms traditional rule-based methods.

The proposed solution contributes to the development of efficient and effective cybersecurity mechanisms for smart buildings, enhancing their security and resilience against cyber threats. The generic nature of the proposed approach means that it can be applied to various types of smart buildings, making it a versatile solution for improving smart building cybersecurity.

In conclusion, this thesis demonstrates the effectiveness of ML techniques in detecting DDoS attacks in smart buildings. The proposed solution can be used to build secure and resilient smart

buildings, ensuring the safety and privacy of occupants and the efficient operation of the building. This thesis adds to the growing body of research on improving the security of smart buildings, which is becoming increasingly important as smart building technology becomes more widespread.

IMPLICATION OF RESEARCH

This work will be useful to software development associations, explicitly in Pakistan. It will help the product improvement organizations, particularly in Pakistan, with the goal that associations can outline various procedures utilizing this method to make their smart homes more secure.

Keywords: *Smart homes, DDoS attacks, Machine Learning.*

ACKNOWLEDGEMENTS

The path to success is not a solitary journey but rather the result of the collective support and prayers of many. I am deeply grateful to Almighty Allah for endowing me with the qualities of perseverance and determination that have enabled me to achieve my goals. His spiritual guidance has been my constant companion, leading me through each step of my journey and illuminating the path ahead. Undoubtedly, I would not have achieved anything without the help and support of the All-Powerful. I am forever indebted to Him for His grace and blessings.

I would also like to express my heartfelt appreciation to my thesis supervisor, Assistant Professor Dr. Yawar Abbas Bangash, for his unwavering support and guidance throughout my thesis. His knowledge, expertise, and dedication to his field have been a source of inspiration to me, and I am grateful for the time and effort he invested in my success. Whenever I encountered any difficulties, he was always available to offer his assistance and provide me with insightful feedback.

In addition, I extend my gratitude to my GEC member, Lt Col. Khawir Mehmood, for his continuous availability for assistance and support throughout my degree, both in coursework and thesis. His expertise and knowledge have been invaluable to me, and I am grateful for his unwavering support and guidance.

Lastly, I would like to express my profound appreciation to my family for their unwavering encouragement and support since day one. Their patience, love, and unwavering support during the challenging times have been a source of strength for me, and I cannot thank them enough. Their unwavering faith in me and their constant support have been instrumental in my success, and I am forever grateful for their presence in my life.

DEDICATION

To my parents, for their unwavering love, guidance, and support that have been instrumental in my academic success.

To my professors and mentors, for their invaluable guidance, expertise, and encouragement to think critically and pursue excellence in my field of study.

To my dear friends, for their support and encouragement that has kept me motivated throughout my academic journey.

To my beloved husband, for his unwavering support, patience, and belief in my abilities that have been my constant source of strength.

Table of Contents

Table of Figures

CHAPTER 1

INTRODUCTION

1.1 Overview:

Internet of things has become an integral part of our daily lives these days. Every upcoming technology has taken account of the Internet of things to provide better and efficient experience to its users. Briefly, it is a combination of different devices (such as sensors, control systems embedded systems and many others) connected with each other via the internet so that they may share and process information. These IoT (Internet of Things) devices can be seen all over the place these days such as smart kitchen appliances, security systems, retail industry, automotive industry, smart building and many more.

Smart building is an environment in which they use automated processes to control the different features in the building such as heating, cooling, ventilation, lighting and many other such features. This automation helps to manage and improve reliability, performance and security of the system.

Besides providing a vast experience and ease in our daily lives, these systems may be prone to network attacks such as Distributed Denial of Service (DDoS) attacks which not only halt the working of the system, as well as may result in data stealing and tempering which can be a huge loss to the stakeholders.

General IoT flow diagram is mentioned below:

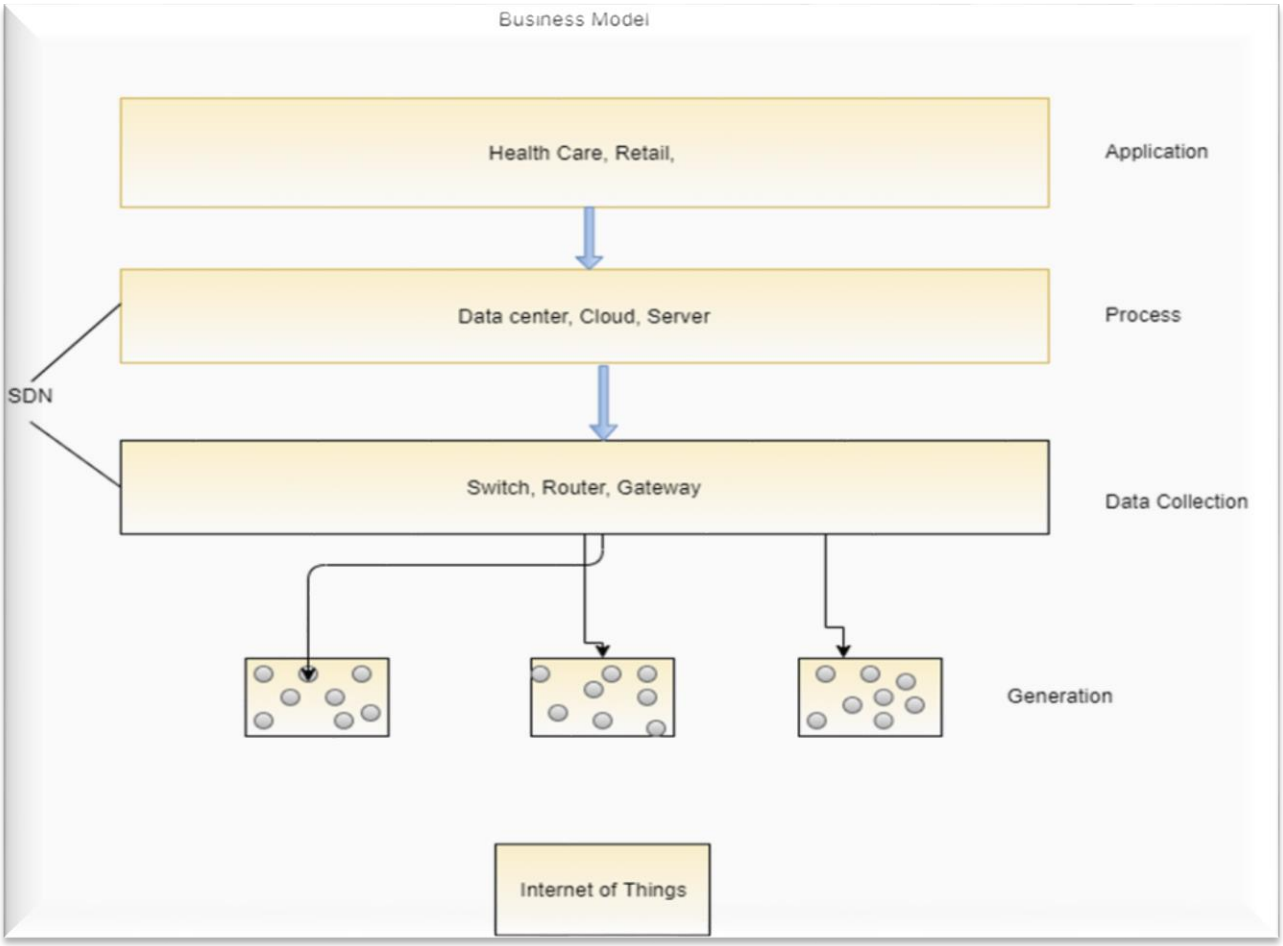


Figure 1

This figure shows that IoT devices are connected to the routers via gateway. These routers are then connected with the SDN device which controllers all the devices under it.

This thesis mainly focuses upon smart buildings as a part of IoT. Smart buildings, enabled by the widespread use of Internet of Things (IoT) devices, are becoming increasingly prevalent. However, this rise in IoT adoption also brings new security challenges, with smart buildings more vulnerable to cyber-attacks, including distributed denial-of-service (DDoS) attacks. DDoS attacks can cause significant damage to the building's network infrastructure, leading to financial losses and downtime.

This thesis proposes a machine learning (ML) based approach to detect DDoS attacks in smart buildings. The proposed solution employs various ML algorithms, including SVM, decision trees, Neural Network using TensorFlow and linear regression. These models are trained to analyse network traffic data collected from smart building devices and detect and classify network traffic patterns that indicate DDoS attacks.

To train the ML models, network traffic data collected from smart buildings is pre-processed to extract relevant features. The performance of the models is evaluated based on accuracy, precision, and recall metrics. The results show that the proposed ML-based approach outperforms traditional rule-based methods.

The proposed solution contributes to the development of efficient and effective cybersecurity mechanisms for smart buildings, enhancing their security and resilience against cyber threats. The generic nature of the proposed approach means that it can be applied to various types of smart buildings, making it a versatile solution for improving smart building cybersecurity.

1.2 Objectives

The objectives of this research are to:

1. Understand the existing available literature and highlight shortcomings
2. Propose a DDoS attacks detection scheme in smart building systems.
3. Propose a mitigation technique to provide secure environment.

To analyze the security and performance of proposed technique

1.3 Relevance to National needs

These days every business and organization, whether public or private, have shifted online and have started to automate their entire environment such as smart building. These automated systems may be prone to different network attacks such as DDoS attacks. These attacks choke the network which may result as a threat for their security too. Thus, to detect such attacks and mitigate these is very important.

1.4 Area of Application

- Security services
- Network intelligence and monitoring applications
- Smart city IoT applications
- E-Health applications etc.

1.5 Advantages

This method will help to avoid DDoS attacks in automated systems such as smart buildings in this way much trustworthy online services may be provided to the customers. At the same time data can also be kept secure without danger of being stolen or tempered.

1.6 Reason/Justification for The Selection of The Topic

These days every business and organization, whether public or private, have shifted online and have started to automate their entire environment such as smart building. These automated systems may be prone to different network attacks such as DDoS attacks. These attacks choke the network which may result as a threat for their security too. Thus, to detect such attacks and mitigate these is very important.

1.7 Thesis Organization

Thesis organization refers to the arrangement and structure of various sections and chapters that compose a thesis or dissertation.

This thesis is formatted in following flow:

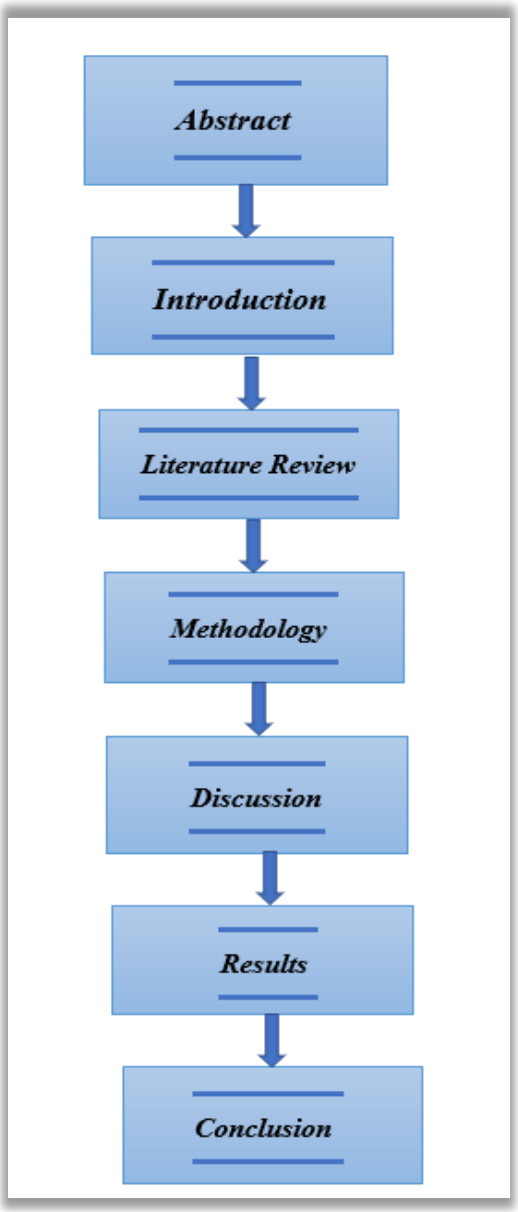


Figure 2

Taxonomy of the thesis

The Internet of Things (IoT) has become ubiquitous, with smart buildings being one of its most common implementations. However, this also makes them more vulnerable to cyber-attacks, particularly Distributed Denial of Service (DDoS) attacks. This thesis proposes a machine learning-based approach to detect DDoS attacks in smart buildings using various ML algorithms, including SVM, decision trees, and linear regression. The proposed solution

trains ML models using pre-processed network traffic data to extract relevant features and detect network traffic patterns indicating DDoS attacks. The approach outperforms traditional rule-based methods, contributing to the development of efficient cybersecurity mechanisms for smart buildings, enhancing their security and resilience against cyber threats.

CHAPTER 2

LITERATURE REVIEW

A literature review is a crucial part of academic research, which involves a critical analysis of existing literature on a specific topic or research question. It requires a comprehensive evaluation, synthesis, and analysis of scholarly works to provide an overview of the current state of knowledge in the field.

The primary purpose of conducting a literature review is to identify significant themes, trends, and findings in the existing research, which researchers can utilize to inform the development of new research questions or hypotheses.

In addition to the types of literature reviews, there are various approaches to conducting a literature review. A deductive approach involves testing a theory or hypothesis, while an inductive approach involves generating new hypotheses or theories based on the literature. A thematic approach involves identifying themes or concepts that emerge from the literature, while a theoretical approach involves using existing theories to frame the review.

Conducting a literature review requires rigorous research skills and a critical mindset. Researchers must identify relevant sources, evaluate the quality of the studies, synthesize the findings, and draw meaningful conclusions. It's also vital to ensure that the review is comprehensive, unbiased, and transparent.

A well-conducted literature review can provide numerous benefits to academic research, such as identifying the research question, defining the problem, selecting appropriate research methods, and highlighting the significance of the study. It also helps to identify gaps in the existing literature and develop new research questions or hypotheses.

Following research papers were thoroughly studied in order to conduct the literature survey.

This literature review provides an overview of research papers that focus on using machine learning algorithms to detect Distributed Denial of Service (DDoS) attacks in smart buildings and IoT devices.

The articles reviewed in this literature review focus on various machine learning techniques used for detecting Distributed Denial-of-Service (DDoS) attacks in different network scenarios. The studies evaluate the effectiveness of different machine learning algorithms and approaches to identify DDoS attacks accurately.

The studies cover a variety of methods, including TRNSYS model, Model Driven Engineering, Fuzzy Logic, Entropy-based method, MQTT protocol, Decision Tree, Random Forest, Naïve Bayes, Support Vector Machine, Linear Regression, K-NN, and Deep Learning methods such as CNN, DNN, AE-SVM, and CNN-LSTM.

Authors in [1] propose a TRNSYS model that assesses the cybersecurity aspect of smart buildings by analyzing HVAC systems. The paper describes the development of a 12-zone HVAC system model and the implementation of a data-driven attack detection framework to detect potential cyber-attacks. The study aims to provide a secure and reliable operation of smart buildings through the detection of attacks in real-time. The research paper was published in 2021. [2] introduces a Model Driven Engineering approach to identify cascading attacks, including DDoS, in smart building systems. It proposes a method for detecting and analyzing cascading attacks in smart building systems using a Model Driven Engineering method and Systems-of-Systems Security (SoSSec) approach. The method is tested in a case study, and the results show that it is able to identify cascading attacks that consist of multiple individual attacks, including Denial-of-Service attacks. The paper was published in 2019.

[3] provides an overview of autonomous machine learning applications in smart buildings. It is a survey that presents an overview of autonomous machine learning applications in smart

buildings. It discusses the learning ability of buildings from a system-level perspective and highlights the benefits of using machine learning in smart building systems. It was published in 2022. In [4] the authors propose a new method to identify DDoS attack patterns on SCADA systems, using machine learning algorithms. The Random Forest classifier achieved a 99.99% accuracy rate for classification. The research paper proposes a new approach for detecting DDoS attack patterns on SCADA systems using machine learning algorithms such as J48, Naive Bayes, Random Forest, and Support Vector Machines. The study reveals that the Random Forest classifier has the highest accuracy rate of 99.99% among all the classifiers. On the other hand, Naive Bayes classifier has the lowest accuracy rate of 97.74%. The study concludes that the proposed approach is effective in detecting DDoS attack patterns on SCADA systems. The [5] paper uses Fuzzy Logic and Machine Learning algorithms to detect Reduction-of-Quality DDoS attacks, achieving a high F1-score. It proposes a method that uses a Multi-Layer Perceptron (MLP) neural network with backpropagation along with three machine learning algorithms, namely K-NN, SVM, and MNB, for the detection of Reduction-of-Quality (RoQ) DDoS attacks. The authors achieved a high F1-score of 98.80% for attack traffic and 99.60% for legitimate traffic, while they achieved a perfect F1-score of 100% for both attack and legitimate traffic for real traffic. The types of DDoS attacks detected using this method include SYN flood attack, Smurf attack, UDP (User Datagram Protocol) flood attack, and DNS flood attack. The paper was published in 2021. In [6], the authors propose a fog layer-based DDoS attack detection approach for IoT devices using clustering and an entropy-based method. The approach uses a clustering and entropy-based method where incoming packets are grouped based on the source IP address, and the overall entropy is calculated to detect DDoS attack traffic. The paper does not mention the types of DDoS attacks detected, and it was published in 2021.

The authors in [7] propose using a message queuing telemetry transport (MQTT) protocol to detect DDoS attacks on IoT devices. It uses a machine learning-based approach for detecting flooding attacks in the context of the Message Queuing Telemetry Transport (MQTT) protocol. The authors conducted experiments that demonstrate how attackers can still overload server resources even if legitimate access to MQTT brokers has been denied and resources have been restricted. The proposed approach achieved an accuracy rate of 99.94% and could be effective in mitigating such attacks in the IoT environment. The paper was published in 2020. [8] introduces an injection attack detection approach using decision tree classifiers, SVM, and Random Forest. The study uses decision tree classifier, support vector machine (SVM), and random forest algorithms to classify the incoming traffic as an attack or normal traffic. The results of the study show that the proposed method achieves a detection rate of 99% for DDOS attacks, including ChopChop, fragmentation, and ARP attacks. The study concludes that the proposed approach can effectively detect injection attacks in smart IoT applications and can provide efficient security measures to protect IoT networks from such attacks. The research was published in 2022. The authors in [9] provide a systematic review of machine learning techniques to detect DDoS attacks in SDN, where several techniques relying on CNN, DNN, AE-SVM, and CNN-LSTM achieved accuracy rates of more than 99%. Specifically, methods based on ResNet, LSTM, DNN, and GRU achieved an accuracy of over 99%. The paper was published in 2023. [10] proposes mathematical models and machine learning models, such as logistic and naïve Bayes, to detect DDoS attacks, achieving 100% accuracy for the machine learning model and 99.75% for the mathematical model. In [11], the authors propose a Machine-Learning-Enabled approach to detect Distributed Denial-of-Service (DDoS) attacks in P4 programmable networks by comparing two DDoS attack detection (DAD) architectures, namely, Standalone and Correlated DAD. The study evaluated the performance of several machine learning algorithms and found that

they achieved high accuracy, precision, recall, and F1-score rates above 98% in most cases. The use of the P4 language to extract features at the data plane reduces latency in real-time DAD implementation, suggesting its effectiveness in detecting DDoS attacks. In [12], J48, Random Forest, and Naïve Bayes algorithms are proposed to detect DDoS attacks, where J48 provided much more efficient results than Random Forest and Naïve Bayes algorithms. The J48 algorithm showed higher accuracy, precision, and recall rates in detecting DDoS attacks. The study was published in 2020, and it provides valuable insights into the application of machine learning algorithms for the detection of DDoS attacks. Overall, the research highlights the potential of machine learning techniques in enhancing the security of network systems against cyber threats such as DDoS attacks.

The [13] proposes a system for detecting DDoS attacks in cloud environments using machine learning algorithms. The study utilizes five popular algorithms: Linear Regression (LR), Support Vector Machines (SVM) with linear, RBF or polynomial kernels, Decision Tree, Naive Bayes, and Random Forest, to detect DDoS attacks from the source side in the cloud. The research findings indicate that the proposed system exhibits a high level of accuracy in detecting attacks, with a success rate of 99.7%. The system also has a low rate of false positives, with less than 0.07% of such incidents occurring. The study focused on detecting three types of DDoS attacks, namely DNS services, HTTP services, and FTP services. The study was published in 2021 and provides valuable insights into the application of machine learning algorithms for detecting DDoS attacks in cloud environments. Overall, the research highlights the potential of machine learning techniques in enhancing the security of cloud systems against cyber threats such as DDoS attacks.

The [14] paper proposes a machine learning-based approach for detecting DDoS attacks using dimensionality reduction techniques. The study uses six machine learning algorithms: Logistic Regression, Decision Tree, KNN, Random Forest Machine Learning Model, SVM,

and NBC. The research findings indicate that when using a smaller dataset, the proposed model demonstrates a decrease in both false negatives and false positives compared to the model trained on the complete dataset. [15] proposes a detection model for DDoS attacks in the web application layer using a semi-supervised learning approach. The study utilizes two algorithms: spectral clustering and random forest to develop the proposed model. The research findings indicate that the proposed model outperforms other detection schemes in terms of effectiveness. The performance of the proposed model is evaluated and compared with other detection schemes to demonstrate its effectiveness in detecting DDoS attacks. The study was published in 2021 and provides valuable insights into the application of semi-supervised learning approaches in the context of DDoS attack detection in web applications. [16] proposes a detection system for DDoS attacks in Software Defined Networking (SDN) environments using machine learning techniques. The study utilizes the random forest algorithm to develop the proposed system. The research findings indicate that several feature selection techniques for machine learning in DDoS detection have been evaluated in this study. The selection of appropriate features is crucial for the classification accuracy of machine learning methods and the efficiency of the SDN controller. Additionally, a comparative examination of feature selection and machine learning classifiers is conducted to identify SDN attacks. The study was published in 2022, and it provides valuable insights into the application of machine learning techniques in the context of DDoS attack detection in SDN environments. In summary, the research highlights the potential of machine learning techniques for enhancing the security of SDN systems against cyber threats such as DDoS attacks. In [17] authors indicate a simulation-based approach to detect DDoS attacks in SDN environments using machine learning algorithms. The study utilizes the SVM, Naïve Bayes, and multi-layered perceptron algorithms to develop the proposed system.

The research findings indicate that the simulation dataset was used to evaluate the classification accuracy of the multilayer perceptron algorithm, which demonstrated the highest accuracy of 99.75% in classifying the traffic. The study highlights the potential of machine learning techniques in the context of SDN-based DDoS attack detection. The study was published in 2021 and provides valuable insights into the application of machine learning algorithms in the context of SDN-based DDoS attack detection. Overall, the research highlights the potential of simulation-based approaches for enhancing the security of SDN systems against cyber threats such as DDoS attacks. [18] proposes a machine learning-based approach to detect DDoS attacks in web servers. The study utilizes various machine learning algorithms, with the Support Vector Machine (SVM) being the primary algorithm for developing the proposed system. The research findings indicate that the proposed approach effectively detects attacks, ensuring uninterrupted services from web servers. The outcomes of the study indicate that the SVM algorithm accurately detects 97.1% of DDoS attacks, surpassing the precision of several existing machine learning approaches. However, the types of DDoS attacks detected were not mentioned in the research paper. The study was published in 2022 and provides valuable insights into the application of machine learning algorithms in the context of DDoS attack detection. Overall, the research highlights the potential of machine learning techniques for enhancing the security of web servers against cyber threats such as DDoS attacks. [19] indicates a method for detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset. The authors use a decision tree classifier and SVM to detect DDoS attacks. The study acknowledges that implementing and testing DDoS strategies can be challenging due to various factors such as complexities, inflexibility, expenses, and vendor-specific architectures of existing networking equipment and protocols. The paper was published in 2019. [20] proposes an open-source, real-time, and robust web application for predicting DDoS attacks, which can be utilized by small to mid-

scale industries to safeguard their networks and servers against harmful DDoS attacks. The proposed approach employs various machine learning algorithms such as Linear Regression, Support Vector Machine, Decision Tree, Naive Bayes, and Random Forest to detect DDoS attacks. The year of publication is 2020. [21] describes the use of three machine learning algorithms, Multilayer Perceptron (MLP), Naïve Bayes, and Random Forest, to detect Distributed Denial of Service (DDoS) attacks. The study used the network simulator NS2 to generate accurate outcomes that reflect real-world conditions, as it is known to produce valid results with high confidence. [22] provides a survey of published papers that use deep learning (DL) techniques to detect DDoS attacks in software-defined networking (SDN). The study focuses on three categories of DL: discriminative, generative, and hybrid learning. The paper discusses the detection of various types of DDoS attacks, including volume-based attacks, protocol attacks, and application plane attacks. The research was published in 2023.

[23] evaluates the effectiveness of three machine learning (ML) algorithms, Decision Tree, Support Vector Machine (SVM), and Convolutional Neural Network (CNN), for detecting intrusions in IoT networks. The study involved analyzing twenty different articles and comparing the ML techniques, datasets, feature engineering methods, and performance indicators used. The paper specifically mentions the detection of SYN flood attacks, Smurf attacks, UDP flood attacks, and DNS flood attacks. The research was published in 2022. [24] uses Support Vector Machine (SVM) to detect DDoS attacks. The study analyzes the attributes "Flow ID," "SYN Flag Cnt," and "Dst IP" and finds that they have the greatest influence on the detection of attacks. The machine learning model was successful in identifying and classifying DDoS attacks with accuracy rates that approached 100%. The research was published in 2020.

[25] uses Decision Tree as a machine learning method to detect denial-of-service (DDoS) attacks in IoT networks. The study assesses the use of deep learning-based intrusion detection

systems (IDS) for meta innovations. The paper indicates that BiLSTMs (Bidirectional Long Short-Term Memory) are more effective for binary classification to distinguish between regular and attacker instances. However, for multiclass classifiers that detect particularly vicious attacks, sequential models such as LSTMs (Long Short-Term Memory) or BiLSTMs yielded superior results. The research was published in 2022

This all information is presented in tabular form so that it may be easily understood and comprehended. Comprehensive literature review is presented in table 1 below:

Table 1: Comprehensive Literature Review

Serial No	Name of Research Paper	Method Used	Observations	Year of publication	Types of DDOS attacks detected
1.	Application of data-driven attack detection framework for secure operation in smart buildings	Transient System Simulation Tool (TRNSYS) model	This paper presents a Transient System Simulation Tool (TRNSYS) model of a 12-zone HVAC system that allows assessing the cybersecurity aspect of HVAC systems.	2021	Not mentioned
2.	Modeling, Analyzing and Predicting Security Cascading Attacks in Smart Buildings Systems-of-Systems	Model Driven Engineering method, Systems-of-Systems Security (SoSSec)	The results from this case study demonstrate that the proposed method discovers cascading attacks comprising of a number of individual attacks, such as a	2019	Not mentioned

			Denial-of-Service attacks		
3.	An overview of machine learning applications for smart buildings	Survey	This review article discusses the learning ability of buildings with a system-level perspective and presents an overview of autonomous machine learning applications	2022	Not mentioned
4.	New Approach to Determine DDoS Attack Patterns on SCADA System Using Machine Learning	J48, Naive Bayes, Random Forest, Support Vector Machines	Results showed that the best classification is obtained using Random Forest classifier (RF) with 99.99% accuracy rate, while Naïve Bayes classifier has the lowest accuracy rate of 97.74%.	2019	Back, Smurf, Neptune, teardrop, Pod, Land
5.	Detection of Reduction-of-Quality DDoS Attacks Using Fuzzy Logic and Machine Learning Algorithms	Multi-Layer Perceptron (MLP) neural network with backpropagation-Nearest Neighbors (K-NN), Support Vector Machine (SVM) and Multinomial Naive Bayes (MNB)	They obtained a F1-score of 98.80% for attack traffic and 99.60% for legitimate traffic, while, for real traffic, they obtained a F1-score of 100% for attack traffic and 100% for legitimate traffic.	2021	SYN flood attack, Smurf attack, UDP (User Datagram Protocol) flood attack, DNS flood attack
6.	Fog Layer-based DDoS attack Detection Approach for Internet-of-Things (IoTs) devices	clustering and entropy-based method	This approach fog node uses the entropy variation in the source's IP address to detect DDoS attack traffic. All the incoming	2021	Not mentioned

			packets are grouped according to the source's IP address and the overall entropy is calculated and compared with a predefined threshold value		
7.	Denial of service attack detection through machine learning for the IoT	Message Queuing Telemetry Transport (MQTT) protocol	The results obtained indicate that the attackers can overwhelm the server resources even when legitimate access was denied to MQTT brokers and resources have been restricted	2020	Flooding attacks
8.	Injection attack detection using machine learning for smart IoT applications	decision tree classifier, SVM and Random Forest	By using this technique, 99% of the DDOS attacks were detected successfully.	2022	ChopChop, fragmentation, and ARP attacks
9.	Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review	Systematic Review	The CICDoS2019 dataset was evaluated using various machine learning and deep learning-based methods to detect DDoS attacks. Results showed that several techniques relying on CNN, DNN, AE-SVM, and CNN-LSTM achieved accuracy rates of more than 99%. Figure 5b presents a comparison of	2023	Not Mentioned

			the performance of these methods. Specifically, methods based on ResNet, LSTM, DNN, and GRU achieved an accuracy of over 99%.		
10.	Detecting Denial of Service attacks using machine learning algorithms	Mathematical model and ML model such as logistic and naive bayes	The accuracy of the machine learning model is 100% and the mathematical model is 99.75%.	2022	Not Mentioned
11.	Machine-Learning-Enabled DDoS Attacks Detection in P4 Programmable Networks	Comparison of two different DAD architectures, called Standalone and Correlated DAD, was done	The results of numerical simulations indicate that the tested machine learning algorithms achieve high accuracy, precision, recall, and F1-score, with rates above 98% in the majority of cases. Moreover, the classification time for most algorithms is within a few hundred microseconds, even in the worst-case scenario. These findings suggest that the use of P4 language to extract features at the data plane can significantly reduce latency in	2021	Not Mentioned

			real-time DAD implementation.		
12.	Detection of DDoS Attacks using Machine Learning Algorithms	J48, Random Forest and Naïve Bayes algorithms	J48 gave much more efficient results than Random Forest and Naïve Bayes algorithm	2020	HTTP flood and SIDDoS
13.	Machine Learning Based DDoS Attack Detection from Source Side in Cloud	Linear Regression (LR), SVM (with linear, RBF or polynomial kernels), Decision Tree, Naive Bayes and Random Forest algorithms	The proposed system exhibits a high level of accuracy in detecting attacks, with a success rate of 99.7%. The system also has a low rate of false positives, with less than 0.07% of such incidents occurring.	2021	DNS services, HTTP services and FTP services.
14.	Machine Learning with Dimensionality Reduction for DDoS Attack Detection	Logistic Regression, Decision Tree, KNN, Random Forest Machine Learning Model, SVM, NBC	When using a smaller dataset, this model demonstrates a decrease in both false negatives and false positives in comparison to the model that was trained on the complete dataset	2022	Not Mentioned
15.	WEB DDoS Attack Detection Method Based on Semi supervised Learning	Spectral clustering and random forest	The current study suggests a detection model for DDoS attacks in the WEB application layer by merging spectral clustering and random forest in a semi-supervised learning approach. The	2021	Not Mentioned

			performance of this model is then compared to other detection schemes to evaluate its effectiveness		
16.	DDoS Detection in SDN using Machine Learning Techniques	Random forest	Several feature selection techniques for machine learning in DDoS detection have been assessed in this study. The selection of appropriate features is crucial for the classification accuracy of machine learning methods and the efficiency of the SDN controller. Additionally, a comparative examination of feature selection and machine learning classifiers is conducted to identify SDN attacks	2022	SYN flood attack, Smurf attack, UDP (User Datagram Protocol) flood attack, DNS flood attack
17.	Simulation of SDN in Mininet and detection of DDoS attack using machine learning	SVM, Naïve Bayes, and multi-layered perceptron	The simulation dataset was utilized to evaluate the classification accuracy of the multilayer perceptron, which demonstrated the highest accuracy of 99.75% in classifying the traffic.	2021	Not Mentioned

18.	Detection of Distributed Denial of Service Attacks based on Machine Learning Algorithms	SVM	Various machine learning methods have been employed in this paper to detect attacks effectively, ensuring uninterrupted services from web servers. The outcomes of the proposed approach indicate that the Support Vector Machine (SVM) accurately detects 97.1% of DDoS attacks, surpassing the precision of several existing machine learning approaches.	2022	Not Mentioned
19.	Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset	decision tree classifier and SVM	Deploying DDoS strategies for testing and implementation can be challenging due to various factors such as the complexities, inflexibility, expenses, and vendor-specific architectures of existing networking equipment and protocols	2019	Not Mentioned
20.	Distributed Denial of Service Attack Detection Using a Machine Learning Approach	Linear Regression (LR), SVM (with linear, RBF or polynomial kernels), Decision Tree, Naive Bayes and	The objective of this proposed work is to bridge this gap by creating an open-source, real-time, and robust web application	2020	Not Mentioned

		Random Forest algorithms	for predicting DDoS attacks. This web application can be utilized by small to mid-scale industries to safeguard their networks and servers against harmful DDoS attacks.		
21.	Detecting Distributed Denial of Service Attacks Using Data Mining Techniques	Multilayer Perceptron (MLP), Naïve Bayes and Random Forest.	The present study utilized the network simulator NS2 due to its ability to generate accurate outcomes that mirror real-world conditions. NS2 was chosen for its reputation for producing valid results with high confidence.	2020	Smurf, UDP-Flood, HTTP-Flood and SIDDOS
22.	Detecting DDoS Attacks in Software Defined Networks Using Deep Learning Techniques: A Survey	Volume-Based Attacks, Protocol Attacks and Application Plane Attacks	This article explores various published papers that employ deep learning (DL) methods to identify distributed denial-of-service (DDoS) attacks in software-defined networking (SDN). The study contrasts three distinct DL categories: discriminative, generative, and hybrid learning.	2023	Not Mentioned
23.	Machine Learning for IoT	Decision Tree, SVM, CNN	This study involved	2022	SYN flood attack, Smurf attack, UDP

	based networks intrusion detection: a comparative study		evaluating machine learning (ML) methods for detecting intrusions in IoT networks. In order to conduct this comparison, we analyzed twenty different articles and compared the ML learning techniques, datasets, feature engineering methods, and performance indicators that were utilized		(User Datagram Protocol) flood attack, DNS flood attack
24.	Analysis and Detection of DDoS Attacks Using Machine Learning Techniques	SVM	The analysis demonstrated that the attributes "Flow ID," "SYN Flag Cnt," and "Dst IP" had the greatest influence on the detection of attacks. Furthermore, the machine learning models successfully identified and classified distributed denial-of-service (DDoS) attacks with accuracy rates that approached 100%.	2020	Not Mentioned
25.	ML-DDoSnet: IoT Intrusion Detection Based on Denial-of-Service Attacks Using Machine Learning	Decision Tree	The present investigation assesses the use of deep learning-based intrusion detection systems (IDS)	2022	Not Mentioned

	Methods and NSL-KDD		for metainnovations. As indicated by previous evaluations, BiLSTMs proved more effective for binary classification distinguishing between regular and attacker instances. However, for multiclass classifiers that detect particularly vicious attacks, sequential models such as LSTMs or BiLSTMs yielded superior results		

The studies reviewed in this literature cover various types of DDoS attacks, including SYN flood attack, Smurf attack, UDP flood attack, DNS flood attack, Flooding attacks, ChopChop, fragmentation, ARP attacks, HTTP flood, SIDDoS, DNS services, HTTP services, and FTP services. It provides an overview of research papers that focus on using machine learning algorithms to detect Distributed Denial of Service (DDoS) attacks in smart buildings and IoT devices. The studies cover a variety of methods, including TRNSYS model, Model Driven Engineering, Fuzzy Logic, Entropy-based method, MQTT protocol, Decision Tree, Random Forest, Naïve Bayes, Support Vector Machine, Linear Regression, K-NN, and Deep Learning methods such as CNN, DNN, AE-SVM, and CNN-LSTM. The papers reviewed propose various methods to detect and analyze DDoS attacks, such as detecting attack patterns on

SCADA systems, identifying cascading attacks, detecting flooding attacks in MQTT protocol, injection attack detection approach, and detecting DDoS attacks in SDN. The studies evaluate the effectiveness of different machine learning algorithms and approaches to identify DDoS attacks accurately. The proposed approaches have achieved high accuracy rates, proving their effectiveness in detecting various types of DDoS attacks in smart buildings and IoT devices.

CHAPTER 3

METHODOLOGY

As the use of smart buildings becomes more widespread, the risk of Distributed Denial of Service (DDoS) attacks is growing. Smart buildings rely on Internet of Things (IoT) devices, which are often interconnected and accessible through the internet. This connectivity makes them vulnerable to cyber-attacks, including DDoS attacks.

To protect against these threats, smart building owners and operators need to implement effective detection methods. Traditional network security measures such as firewalls and intrusion detection systems may not be sufficient to protect against the increasingly sophisticated attacks.

An effective approach to detecting DDoS attacks is anomaly detection. This method involves monitoring system behaviour for unusual activity, using machine learning algorithms to identify patterns and anomalies in the data. By detecting anomalies in real-time, security teams can quickly respond to potential threats and minimize their impact.

Another effective method is the use of traffic analysis tools that can identify and block traffic from known malicious sources. This can help prevent attacks before they cause significant damage. In conclusion, the rise of DDoS attacks in smart buildings highlights the importance of effective detection methods. Anomaly detection and traffic analysis tools can help mitigate the impact of these attacks and ensure the continued safety and reliability of smart building systems.

This research aims to propose a machine learning (ML) based approach to detect DDoS attacks in smart buildings. The proposed solution employs various ML algorithms, including SVM, decision trees, Neural Network using TensorFlow and linear regression. These models are trained to analyse network traffic data collected from smart building devices and detect and classify network traffic patterns that indicate DDoS attacks.

We carried out an extensive literature review which helped us to understand various methods and techniques used to detect DDOS attacks in the Smart Building system.

As we are using dataset which is generated by the simulator, we think that there might be some limitations. As we know that simulated data is frequently used in cybersecurity to train machine learning models, including for detecting DDoS attacks in smart buildings. However, while simulated data can be useful in some contexts, it also has several limitations that can impact the effectiveness of the model in real-world scenarios.

One limitation of simulated data is that it may not accurately represent the diversity of real-world data. The data used to generate simulations may not fully capture the complexity and unpredictability of actual network traffic, leading to oversimplified models that are not effective in identifying new types of attacks.

Another limitation of simulated data is that it may not accurately reflect the distribution of traffic in real-world environments. Simulated data may not capture the variability in network traffic that can result from factors such as changes in user behaviour or seasonal trends. As a result, models that are poorly calibrated to the real-world environment may generate high false positive or false negative rates.

Finally, the use of simulated data may introduce biases into the model that can affect its performance in detecting real-world attacks. For example, the simulated data may be biased towards certain types of attacks or network configurations, resulting in a less effective model in detecting attacks in other contexts.

By keeping in mind all of above concerns, we have, very carefully, crafted our methodology which has given us very promising results.

3.1 Proposed Algorithm

In this section, an algorithm for detecting DDoS attacks will be presented. The aim of this research is to propose a method to reduce data dimensions by identifying the key features that have the greatest impact on forecast accuracy.

Following steps were taken in formulation of this algorithm:

1. Selection of base paper
2. Choosing dataset
3. Cleaning of data
4. Data pre-processing
5. Data split into two parts (test data and train data)
6. Application of ML techniques (Decision tree, Neural Network using TensorFlow, SVM and Linear Regression)
7. Accuracy calculation of the ML Techniques

This all is shown in a diagram below:

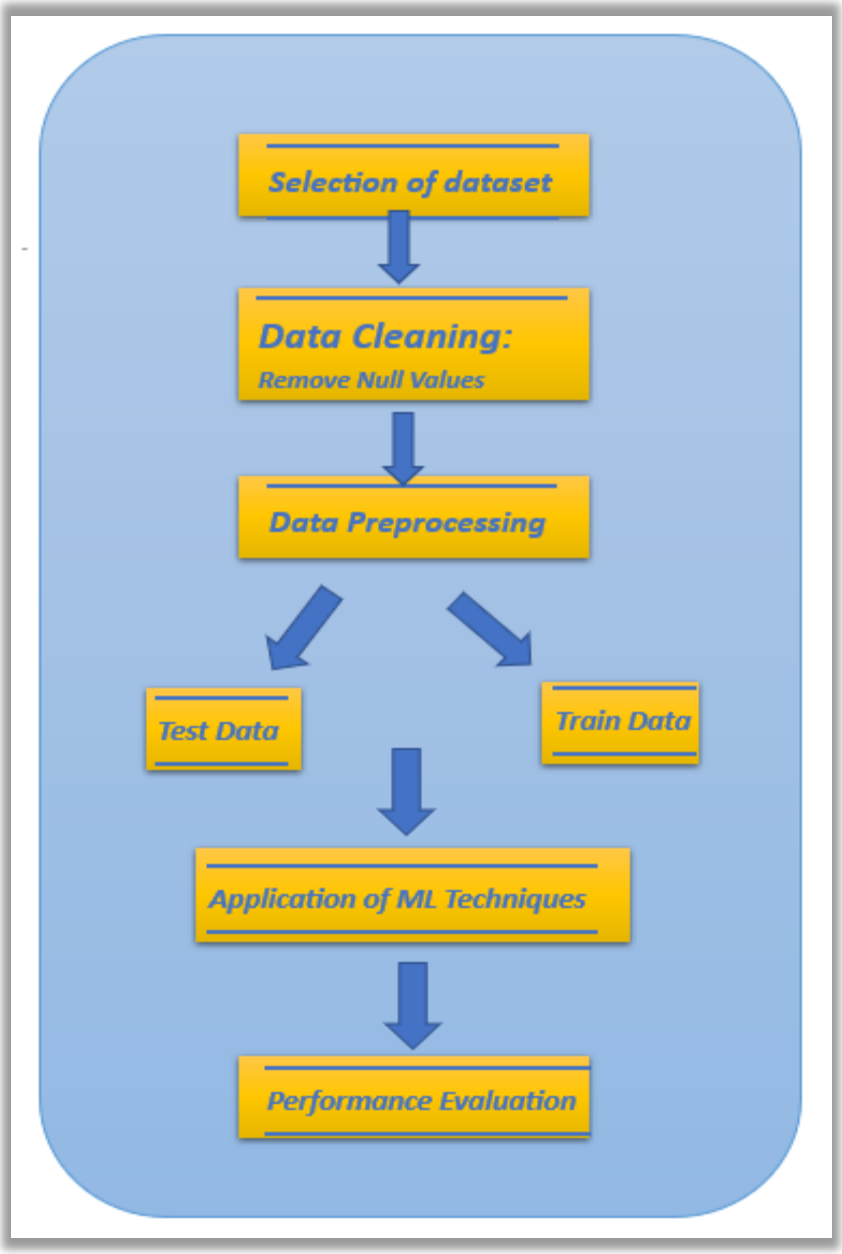


Figure 3

Our proposed algorithm starts with selection of dataset. This dataset was then cleaned. Then data preprocessing was done and data was divided into test and train data. Then Machine learning techniques were applied and performance evaluation was done.

Our Dataset is the Simulated data collected from the HVAC system of the smart building. Data collection was enabled by the sensors used in HVAC system. This smart building is divided into four zones

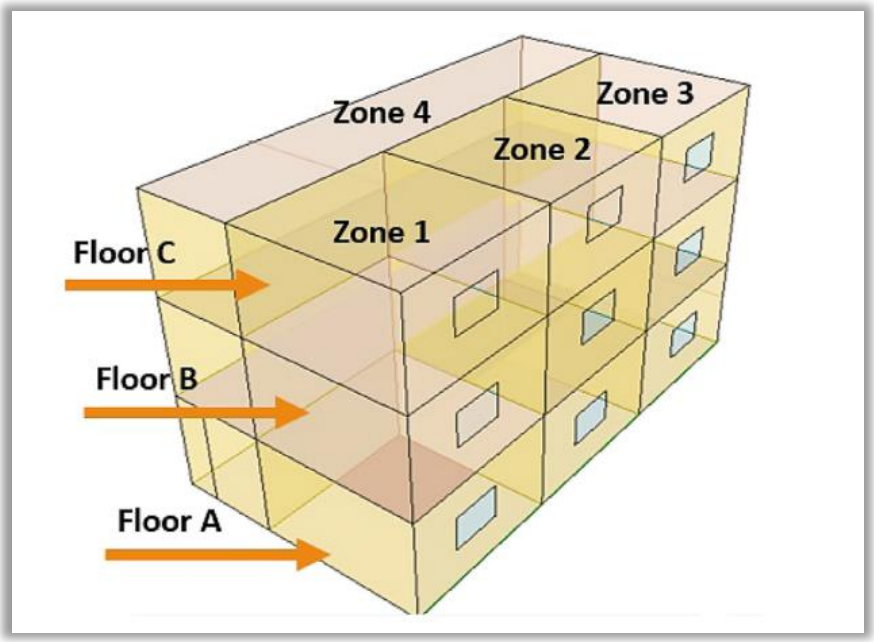


Figure 4

This diagram is a sketch of the smart building which is divided into 3 floors and 4 zones connected to a HVAC system. Data from sensors of HVAC system was captured and noted with respect to time. [1]

All of these zones are equipped with Heating, Cooling, Ventilation and Airconditioning (HVAC) system and data is being monitored with the help of the sensors. All this data is being collected with respect to time.

Here is a glimpse of dataset that we used.

	Hour of the year	ur of the	Temperature sensors measurements																					
			Tamb	Floor A				Floor B				Floor C				AHU						Chiller System		Floc	
				Tz1	Tz2	Tz3	Tz4	Tz1	Tz2	Tz3	Tz4	Tz1	Tz2	Tz3	Tz4	T_aoA	T_aoB	T_aoC	T_woA	T_woB	T_woC	T_t	T_chiller	Uz1	Uz2
4	5094.791	6.791	28.227	20.197	21.169	18.528	20.274	21.746	20.358	20.229	19.395	20.151	21.177	20.157	18.439	12.999	12.169	12.164	13.168	13.37	13.36	10.898	8.999	0.214	0.24
5	5094.806	6.806	28.239	20.061	21.085	18.457	20.206	22.614	20.271	19.972	19.325	19.993	21.086	19.978	18.385	12.996	12.201	12.192	13.197	13.396	13.38	10.928	8.996	0.2	0.222
6	5094.821	6.821	28.251	20.185	21.182	18.544	20.272	22.274	20.369	20.122	19.392	20.316	21.201	20.305	18.448	13.009	12.212	12.204	13.203	13.404	13.388	10.986	9.008	0.159	0.197
7	5094.836	6.836	28.263	20.155	21.12	18.468	20.195	21.22	20.303	20.2	19.312	20.27	21.143	20.282	18.361	12.994	12.143	12.141	13.132	13.334	13.328	10.863	8.993	0.197	0.227
8	5094.851	6.851	28.275	20.087	21.087	18.437	20.176	21.521	20.267	20.08	19.291	20.005	21.096	20.009	18.338	12.989	12.202	12.197	13.193	13.381	13.372	10.919	8.989	0.189	0.209
9	5094.866	6.866	28.287	20.273	21.276	18.622	20.364	22.621	20.46	20.191	19.48	20.261	21.291	20.243	18.521	13.018	12.222	12.215	13.203	13.405	13.39	11.046	9.017	0.169	0.199
10	5094.881	6.881	28.299	20.124	21.133	18.484	20.234	22.179	20.31	20.081	19.348	20.258	21.14	20.251	18.4	13.001	12.112	12.106	13.087	13.299	13.288	10.861	9	0.226	0.257
11	5094.896	6.896	28.31	20.061	21.078	18.441	20.191	21.317	20.252	20.078	19.301	20.095	21.074	20.108	18.365	13.004	12.196	12.192	13.177	13.367	13.359	10.985	9.004	0.182	0.214
12	5094.91	6.91	28.322	20.131	21.102	18.45	20.182	21.704	20.277	20.13	19.293	20.077	21.118	20.077	18.344	12.997	12.142	12.136	13.111	13.315	13.305	10.891	8.997	0.163	0.198
13	5094.925	6.925	28.334	20.176	21.139	18.465	20.2	22.618	20.313	20.122	19.31	20.234	21.169	20.215	18.346	12.992	12.178	12.171	13.145	13.347	13.332	10.917	8.992	0.185	0.205
14	5094.94	6.94	28.346	20.184	21.202	18.527	20.281	22.295	20.376	20.148	19.392	20.295	21.218	20.292	18.428	13.003	12.187	12.18	13.153	13.352	13.338	10.969	9.002	0.199	0.217
15	5094.955	6.955	28.358	20.169	21.213	18.554	20.32	21.325	20.383	20.17	19.429	20.153	21.207	20.165	18.478	13.016	12.158	12.154	13.121	13.321	13.313	10.993	9.015	0.202	0.236
16	5094.97	6.97	28.37	20.1	21.114	18.471	20.231	21.606	20.279	20.075	19.335	20.051	21.104	20.048	18.4	13.013	12.123	12.118	13.08	13.286	13.276	10.95	9.013	0.198	0.24
17	5094.985	6.985	28.382	20.043	21.016	18.366	20.11	22.487	20.18	19.995	19.212	20.127	21.024	20.11	18.273	12.997	12.123	12.116	13.074	13.282	13.268	10.889	8.996	0.178	0.211
18	5095	7	28.394	20.08	21.05	18.337	20.096	22.211	20.19	20.072	19.176	20.168	21.061	20.166	18.206	12.989	12.159	12.153	13.11	13.309	13.296	10.906	8.989	0.161	0.182
19	5095.015	7.015	28.406	20.092	21.098	18.326	20.119	21.326	20.207	20.095	19.169	20.035	21.085	20.042	18.166	12.987	12.158	12.152	13.11	13.308	13.296	10.902	8.987	0.173	0.192
20	5095.03	7.03	28.418	20.169	21.196	18.446	20.24	21.704	20.314	20.129	19.289	20.129	21.174	20.12	18.281	13.008	12.185	12.181	13.144	13.335	13.326	11.013	9.008	0.177	0.206
21	5095.045	7.045	28.43	20.001	21.027	18.322	20.109	22.481	20.161	19.943	19.17	20.079	21.005	20.063	18.181	12.993	12.114	12.106	13.068	13.276	13.261	10.855	8.992	0.201	0.237
22	5095.06	7.06	28.442	19.972	20.963	18.255	20.035	22.168	20.096	19.969	19.101	20.023	20.952	20.023	18.122	12.986	12.185	12.179	13.149	13.341	13.328	10.913	8.985	0.15	0.186
23	5095.075	7.075	28.454	20.27	21.215	18.463	20.233	21.494	20.338	20.306	19.297	20.247	21.228	20.252	18.301	13.013	12.219	12.212	13.185	13.377	13.364	11.039	9.012	0.142	0.167
24	5095.09	7.09	28.466	20.108	21.128	18.372	20.161	21.748	20.244	20.051	19.217	20.084	21.119	20.075	18.221	13	12.114	12.109	13.079	13.285	13.275	10.877	8.999	0.232	0.243
25	5095.104	7.104	28.478	19.972	21.059	18.341	20.143	22.578	20.181	19.875	19.2	19.998	21.016	19.983	18.219	13.01	12.204	12.196	13.177	13.372	13.356	11.015	9.009	0.184	0.217
26	5095.119	7.119	28.49	20.019	21.005	18.291	20.076	22.071	20.132	20.052	19.135	20.074	20.985	20.074	18.154	12.999	12.134	12.128	13.104	13.311	13.298	10.887	8.998	0.143	0.196
27	5095.134	7.134	28.501	20.14	21.056	18.309	20.081	21.273	20.177	20.204	19.141	20.16	21.072	20.167	18.147	12.994	12.191	12.188	13.172	13.362	13.355	10.938	8.994	0.157	0.18
28	5095.149	7.149	28.513	20.13	21.136	18.37	20.153	21.751	20.254	20.053	19.212	20.114	21.147	20.104	18.21	12.999	12.193	12.188	13.176	13.369	13.359	10.947	8.998	0.194	0.196
29	5095.164	7.164	28.525	19.997	21.083	18.331	20.101	22.511	20.205	19.903	19.161	20.008	21.061	19.992	18.166	12.986	12.182	12.177	13.169	13.362	13.352	10.872	8.985	0.192	0.22

Attaxk (1) | Attack (2) | Attack (3) | Attack (4) | Attack (5) | Attack (6) | Attack (7) | Attack (8) | ...

Figure 5

Snippet of dataset is attached here which shows the data captured from the sensors with respect to time. [1]

Below is the list of columns that are part of the dataset

```
In [24]: df.columns

Out[24]: Index(['Hour of the year', 'Hour of the day',
                'Temperature sensors measurements ', 'Unnamed: 3', 'Unnamed: 4',
                'Unnamed: 5', 'Unnamed: 6', 'Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9',
                'Unnamed: 10', 'Unnamed: 11', 'Unnamed: 12', 'Unnamed: 13',
                'Unnamed: 14', 'Unnamed: 15', 'Unnamed: 16', 'Unnamed: 17',
                'Unnamed: 18', 'Unnamed: 19', 'Unnamed: 20', 'Unnamed: 21',
                'Unnamed: 22', 'Control signals', 'Unnamed: 24', 'Unnamed: 25',
                'Unnamed: 26', 'Unnamed: 27', 'Unnamed: 28', 'Unnamed: 29',
                'Unnamed: 30', 'Unnamed: 31', 'Unnamed: 32', 'Unnamed: 33',
                'Unnamed: 34', 'Unnamed: 35', 'Setpoints', 'Unnamed: 37', 'Unnamed: 38',
                'Unnamed: 39', 'Unnamed: 40', 'Unnamed: 41', 'Unnamed: 42',
                'Unnamed: 43', 'Unnamed: 44', 'Unnamed: 45', 'Unnamed: 46',
                'Unnamed: 47', 'Unnamed: 48', 'Unnamed: 49', 'Unnamed: 50',
                'Thermal Comfort indices (PMV)', 'Unnamed: 52', 'Unnamed: 53',
                'Unnamed: 54', 'Unnamed: 55', 'Unnamed: 56', 'Unnamed: 57',
                'Unnamed: 58', 'Unnamed: 59', 'Unnamed: 60', 'Unnamed: 61',
                'Unnamed: 62', 'Total power usage', 'Label'],
               dtype='object')
```

Figure 6

List of columns in the dataset. [1]

To elaborate dataset more, following is the list of abbreviations used in it

Table 2: List of abbreviations

Symbols	Subscripts
T	Temperature
U	Control Signal
PMV	Predicted Mean Vote
P	Power
T	Time
Z	Zone
Ao	Output Air
Wo	Output Water
Amb	Ambient
D	Day
Y	Year

Data parameters are mentioned as below

Table 3: List of Data Parameters

Index	Symbol	Description
1	ty	Hour of the year
2	td	Hour of the day
3	Tamb	The ambient temperature (‘C)
4-15	TzA1-TzA4-TzB1-TzB4-TzC1-TzC4	The temperature of the zones (‘C)
16-18	TaoA-TaoB-TaoC	The temperature of Air Handling Unit (AHU) supply air (‘C)
19-21	TwoA-TwoB-TwoC	The temperature of cooling coil return water (‘C)
22	Tt	The temperature of chilled water tank (‘C)
23	Tchiller	The temperature of chiller outlet water (‘C)
24-36	U1-U13	The control signals
37-51	-	The temperature setpoints (‘C)
52-63	PMV1-PMV12	The zones thermal comfort indices
64	Ptotal	The overall estimated power utilization of HVAC system
65	label	The label of the system status

List of attacks detected with respect to time is given below:

Table 4: List of attacks

List of anomalous data that is captured by the sensors of HVAC system. This data is shown with respect to time. [1]

Attack Index	Description	Attack Time
1.1	Changing the setpoint of the chiller to 14’C	Day 1, 12:00
1.2	Changing the setpoint of the water tank to 16’C	Day 2, 06:00
1.3	Changing the setpoint of the AHU to 20’C	Day 2, 10:00
1.4	Changing the setpoint of the Zone A1 to 26’C	Day 20,11:00
1.5	Changing the setpoint of zone C4 to 18’C	Day 1, 03:00
2.1	Freezing Zone B1 reading	Day 5, 16:00
2.2	Freezing Zone C4 reading	Day 7, 06:00
2.3	Freezing Zone A2 reading	Day 9, 04:00
2.4	Freezing Zone C3 reading	Day 10, 06:00
2.5	Introducing a bias of 3’C to Zone B3	Day 3, 06:00
3.1	Freezing the control signal of Zone C2	Day 10, 15:00
3.2	Freezing the control signal of Zone B3	Day 13, 18:00
3.3	Freezing the control signal of Zone B1	Day 15, 06:00
3.4	Setting control signal of Zone B2 to 0	Day 19, 14:00
3.5	Setting control signal of Zone A3 to 1	Day 19, 20:00
4.1	Reducing the AHU-B water pump to 1/3 of its speed	Day 18, 12:00

We have used 4 algorithms which gave us promising outcome. We used Decision Tree, SVM, Linear Regression and Neural Network using TensorFlow. Let us have a look on Pseudocodes of these.

Decision Tree:

```
GenDecTree(Sample S, Features F)
Steps:
1. If stopping_condition(S, F) = true then
    a. Leaf = createNode()
    b. leafLabel = classify(s)
    c. return leaf
2. root = createNode()
3. root.test_condition =findBestSpilt(S,F)
4. V = {v | v a possible outcomecfroot.test_condition}
5. For each value v ∈ V:
    a. S_v = {s | root.test_condition(s) = v and s ∈ S };
    b. Child = TreeGrowth (S_v, F);
    c. Add child as descent of root and label the edge {root → child} as v
6. return root
```

Figure 7

Pseudocode of decision tree algorithm [34]

The initial stage of constructing a decision tree involves iteratively dividing the training dataset using the most effective criteria until the majority of data within each partition share the same class label, potentially leading to overfitting. In the subsequent phase, the branches are pruned to ensure that the resulting tree can generalize well and avoid overfitting.

SVM:

Data : Dataset with p^* variables, time-to-event and status.

Input : Number of equidistant cutoff points c^*.

Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model;

$p \leftarrow p^*$;

while $p \geq 2$ **do**

  $SVM_p \leftarrow$ SVM with the optimized tuning parameters for the p variables and observations in **Data**;

  for $i = 1$ **to** p **do**

    $pseudo_i \leftarrow$ prediction vector of c^* pseudo-samples for variable i;

    $rank.criteria_i \leftarrow$ Median Absolute Deviation of $pseudo_i$ vector;

  end

  $min.rank.criteria \leftarrow$ variable with lowest value in $(rank.criteria_1, \ldots, rank.criteria_p)$;

  Remove $min.rank.criteria$ from **Data**;

  $Rank_p \leftarrow min.rank.criteria$;

  p ← p - 1 ;

end

$Rank_1 \leftarrow$ variable in **Data** $\notin (Rank_2, \ldots, Rank_{p^*})$;

return $(Rank_1, \ldots, Rank_{p^*})$

Figure 8

Pseudocode of SVM algorithm [34]

Linear Regression:

Preliminaries
combine all n LC/MS runs
build overlapping mass-windows across combined runs
1. Cluster Analysis
for each mass-window **do**
use p peaks with highest intensities
calculate distance matrix of pairs of peaks (j, h)

$$d_{j,h} = \begin{cases} \text{diff}(mass), & \text{if } \text{diff}(rt) < k_1 \quad \wedge \\ & \text{diff}(\log_{10}(intensity)) < k_2 \\ \infty, & \text{if } \text{diff}(rt) \geq k_1 \quad \vee \\ & \text{diff}(\log_{10}(intensity)) \geq k_2 \end{cases}$$

hierarchical average linkage cluster analysis
cut cluster-tree at mass accuracy Δ_m
if $n_{dup} < threshold_1 \quad \wedge \quad n_{miss} < threshold_2$ **then**
cluster is 'well-behaved'
delete duplicated 'well-behaved' clusters
for each 'well-behaved' cluster **do**
$\tilde{rt} = median(rt)$
for each peak i **do**
$dev_i = rt_i - \tilde{rt}$
2. Regression
for each run s **do**
take only peaks from 'well-behaved' clusters
fit regression line $\hat{dev}_{s,i} = a_s + b_s * rt_i$
by minimizing $\sum (dev_i - \hat{dev}_{s,i})^2$
Correction
for each run s **do**
for each peak i **do**
$rt_{cor,i} = rt_i - \hat{dev}_{s,i}$

Figure 9

Pseudocode of Linear Regression Algorithm [35]

Figure 9 presents the pseudocode for the linear regression method utilized in the alignment process, which is a two-step procedure. Firstly, cluster analysis is employed to identify groups of peaks that can be easily aligned. Then, in the second step, linear regression is applied to estimate the rt deviation line for each run using these groups, enabling the estimation of global trends.

Neural Network using TensorFlow:

1: Initialise Wi, Ai (word_embedding, attention embedding)
2: while not convergent do
3: for l ∈ {0, . . . , L − 1} do
4: 1) Compute x_cap using eq (i) (rnn network part)
5: x_cap = f rnn (l; θrnn) ∈ R D (where l is the sentence or input , θrnn indicates model parameters and D is the dim)
6: compute forward pass for lstm
7: $\overrightarrow{h}_i = \overrightarrow{\text{LSTM}}(\overrightarrow{h}_{i-1}, x_i)$
8: compute backward pass for lstm
9: $\overleftarrow{h}_i = \overleftarrow{\text{LSTM}}(\overleftarrow{h}_{i+1}, x_i)$
10: $h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i]$,
11: 2) Compute xt using eq (ii) (nano network part)
12: $$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$
13: y_cap = x_cap . α_{ij}
14: end for
15: Calculate Cross-Entropy $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \left[-y_k^{(i)} \log((h_\theta(x^{(i)}))_k - (1 - y_k^{(i)}) \log(1 - (h_\theta(x^{(i)}))_k\right]$
16: loss = reduce_mean(cross_entropy)
17: update the network parameters basis on loss using Back propagation $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$
18: end while
19: Return y_cap

Figure 10

Pseudocode of Neural Network using TensorFlow [35]

Results attained by applying above mentioned methodology is discussed in detail in the Result section.

CHAPTER 5

RESULTS

The results section of a thesis is a critical component that presents the findings of the research study in a comprehensive and meaningful manner. It is an opportunity for the researcher to analyse and interpret the data gathered from various research methods and present them in an organized and structured way. The results section is crucial in demonstrating the validity of the research and the extent to which the research question or hypothesis has been answered.

In conclusion, the results section of a thesis is a critical component that presents the findings of the research study. The researcher should pay attention to the organization of the data, the language used, and the critical evaluation of the results presented. A well-crafted results section will provide insights into the validity of the research and the implications for the field of study.

The methodology that we adopted here for our thesis gave us very promising results. All of it is discussed below in detail.

4.1 Decision Tree:

A decision tree is a machine learning algorithm commonly used for classification and regression tasks. It is a type of supervised learning algorithm that creates a tree-like model of decisions and their possible consequences.

The algorithm works by recursively partitioning the data into subsets based on the values of input features, with the aim of creating homogeneous subgroups that have similar values for the output variable. At each step of the tree-building process, the algorithm selects the feature that provides the most information gain or the best split to partition the data.

To make predictions using the decision tree, one follows a path through the tree based on the values of input features. At each node, a decision is made based on the value of a specific feature, and the prediction is based on the outcome associated with that decision.

One of the benefits of decision trees is that they can handle both categorical and numerical data, and they are easy to interpret. However, they can be prone to overfitting, where the tree is overly complex and captures noise in the data rather than the underlying patterns. To mitigate this issue, various techniques, such as pruning and setting limits on the depth of the tree, can be used.

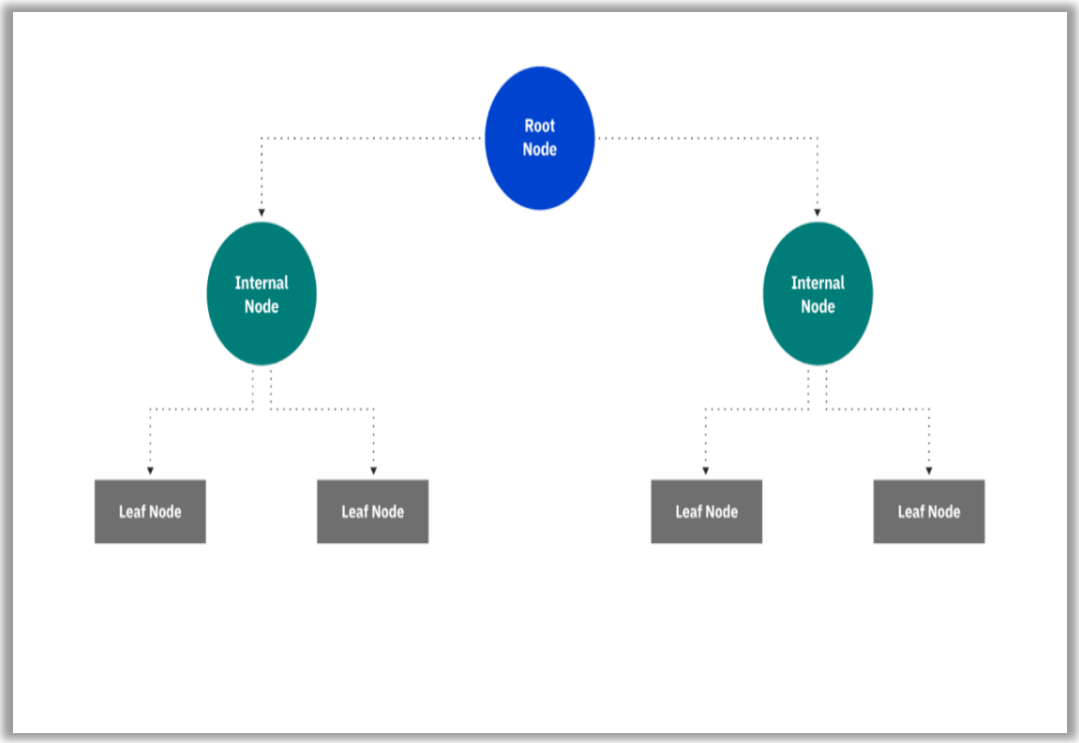


Figure 11

This diagram shows how decision trees are modeled in s graphical form. [29]

We have used Decision Tree here which gave us 100% accuracy. Code for Decision Tree is given as below:

```
In [25]: feature_cols = ['Hour of the year', 'Hour of the day',
               'Temperature sensors measurements ','Total power usage' ]
         X = df[feature_cols] # Features
         y = df.Label
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% training
         # Create Decision Tree classifer object
         clf = DecisionTreeClassifier()

         # Train Decision Tree Classifer
         clf = clf.fit(X_train,y_train)

         #Predict the response for test dataset
         y_pred = clf.predict(X_test)

         # Model Accuracy, how often is the classifier correct?
         print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

         Accuracy: 1.0
```

Figure 12

Code for Decision Tree

4.2 Neural Network using TensorFlow

TensorFlow is a software library that is available for free and open-source use. It is primarily used for creating and training machine learning models, and was developed by the Google Brain team. TensorFlow is designed to be flexible and scalable, allowing users to create deep neural networks and other machine learning algorithms that can handle large amounts of data. It also provides developers with a high-level API for building and training models, as well as a lower-level API for customizing and optimizing models for specific use cases. TensorFlow has become one of the most popular machine learning libraries and is widely used in various fields, including image and speech recognition, natural language processing, and predictive analytics.

Neural networks aim to replicate the human brain's functions and are a type of machine learning model. TensorFlow is a well-known open-source software library used for developing and training neural networks.

To create a neural network in TensorFlow, the data is processed through a series of layers, each with a unique responsibility. The initial layer is the input layer, which receives the raw data, and the output layer is the final layer, generating the ultimate output of the model. Between these two layers are hidden layers that process the data in different ways.

The tf.keras API in TensorFlow is employed to construct neural networks, which provides a simple interface for their creation and training. To build a neural network in TensorFlow, it is essential to specify the number of layers, the quantity of neurons in each layer, and the type of activation function to be used, in order to define the network's architecture.

Diagrammatical view of TensorFlow is given below:

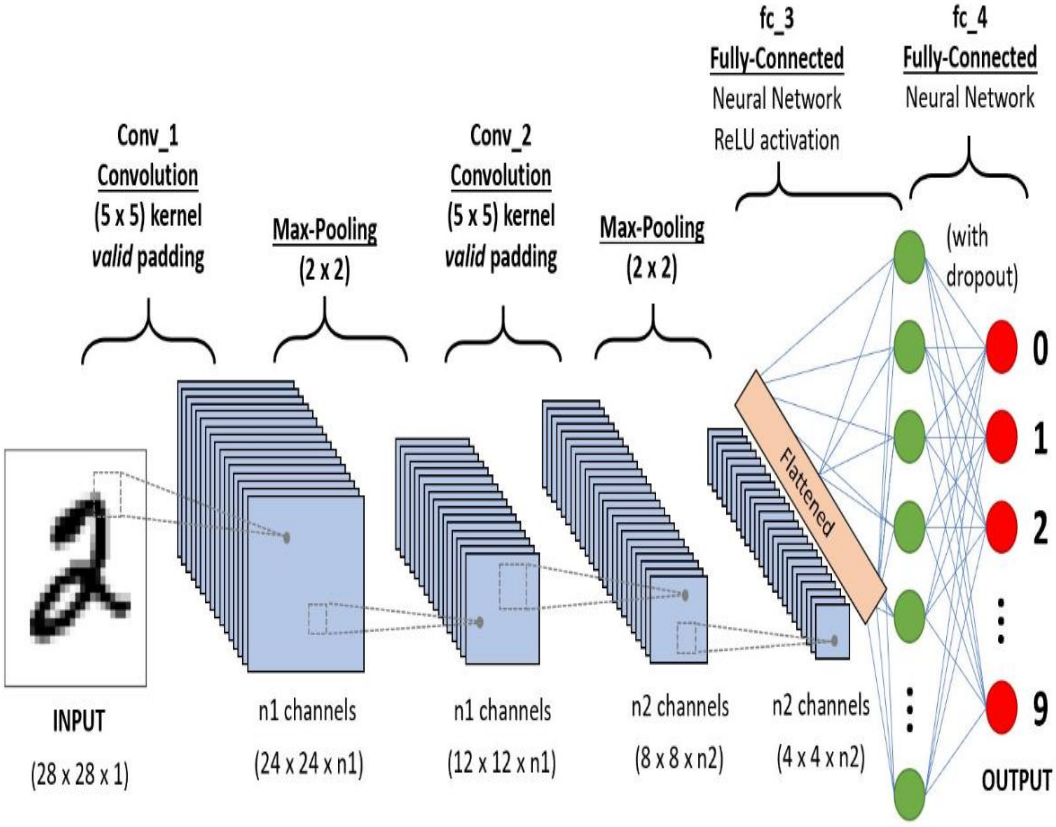


Figure 13

Overall diagram of working of TensorFlow. Here input image is given and neural network output is shown [30]

TensorFlow has various types of layers, such as dense, convolutional, and recurrent layers, which are suitable for processing different types of data. Dense layers link all neurons in one layer to the next, while convolutional layers are ideal for processing images or data with spatial relationships. Recurrent layers are better suited for processing sequences of data, such as text or time series.

Training a neural network in TensorFlow requires defining the model architecture, specifying the loss function, optimizer, and metrics, and feeding it with training data. During training, the model adjusts its weights and biases to reduce the loss function, which calculates the difference between the predicted output and the true output. The optimizer optimizes the

weights and biases to minimize the loss function, while metrics are used to evaluate the model's performance.

After training, the neural network can be utilized to make predictions on new data. TensorFlow provides tools for saving and loading trained models, making it simple to use them in other applications. Due to TensorFlow's versatility and scalability, it is a popular choice for developing and training neural networks in various fields, including computer vision, natural language processing, and speech recognition.

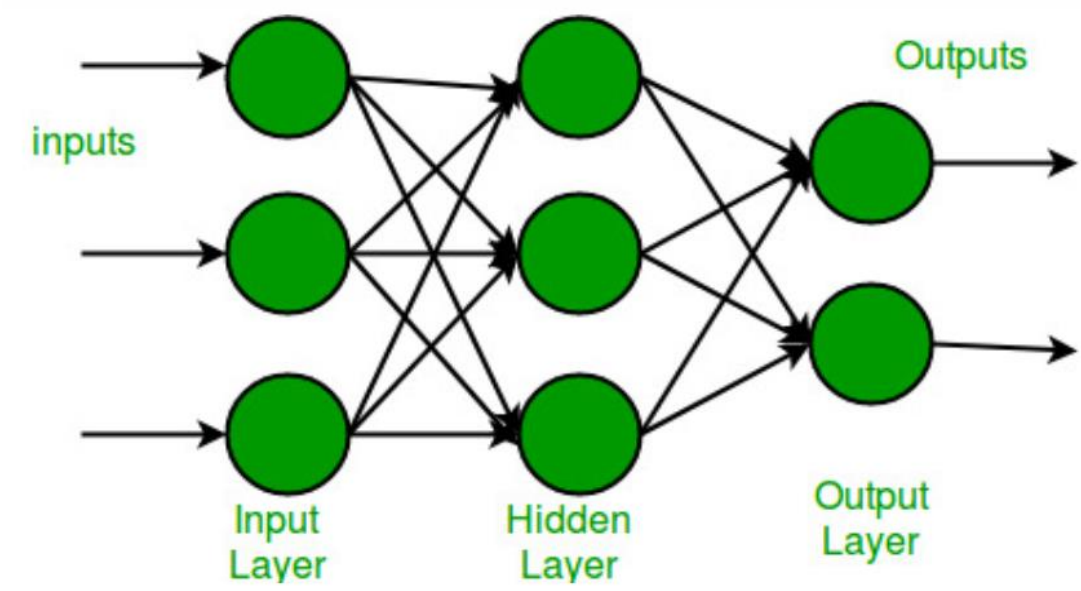


Figure 14

This image shows TensorFlow, input layer, hidden layer and output layer [31]

Detailed code of it is given below:

```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split

In [2]: df= pd.read_csv("C:\\Users\\yumna\\OneDrive\\Desktop\\HVAC system dataset - Log 3 - Final.csv")

In [4]: X = pd.get_dummies(df.drop(['Label'], axis=1))
        y = df['Label']

In [5]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2)

In [6]: y_train.head()

Out[6]: 329    0.0
        247    0.0
        341    0.0
        361    1.0
        328    0.0
        Name: Label, dtype: float64
```

Figure 15

TensorFlow code Part 1

```
In [7]: from tensorflow.keras.models import Sequential, load_model
        from tensorflow.keras.layers import Dense
        from sklearn.metrics import accuracy_score

In [8]: model = Sequential()
        model.add(Dense(units=32, activation='relu', input_dim=len(X_train.columns)))
        model.add(Dense(units=64, activation='relu'))
        model.add(Dense(units=1, activation='sigmoid'))
```

Figure 16

TensorFlow code part 2

```
In [26]: model.fit(X_train, y_train, epochs=200, batch_size=32)
Epoch 192/200
15/15 [==============================] - 0s 5ms/step - loss: nan - accuracy: 0.5844
Epoch 193/200
15/15 [==============================] - 0s 5ms/step - loss: nan - accuracy: 0.5844
Epoch 194/200
15/15 [==============================] - 0s 5ms/step - loss: nan - accuracy: 0.5844
Epoch 195/200
15/15 [==============================] - 0s 5ms/step - loss: nan - accuracy: 0.5844
Epoch 196/200
15/15 [==============================] - 0s 4ms/step - loss: nan - accuracy: 0.5844
Epoch 197/200
15/15 [==============================] - 0s 5ms/step - loss: nan - accuracy: 0.5844
Epoch 198/200
15/15 [==============================] - 0s 5ms/step - loss: nan - accuracy: 0.5844
Epoch 199/200
15/15 [==============================] - 0s 5ms/step - loss: nan - accuracy: 0.5844
Epoch 200/200
15/15 [==============================] - 0s 5ms/step - loss: nan - accuracy: 0.5844

Out[26]: <keras.callbacks.History at 0x22f1830f3d0>
```

Figure 17

TensorFlow code part 3

```
In [27]: y_pred = model.predict(X_test)

         4/4 [==============================] - 0s 7ms/step

In [34]: y_pred = [0 if val < 0.5 else 1 for val in y_hat]

In [35]: print("Accuracy:",accuracy_score(y_hat, y_pred))

         Accuracy: 1.0
```

Figure 18

TensorFlow code part 4

4.3 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are widely-used machine learning algorithms that are commonly used for both classification and regression tasks. SVMs are a type of supervised learning algorithm, meaning that they learn from labeled training data.

The SVM algorithm works by finding the optimal hyperplane that separates the data points into different classes. The optimal hyperplane is chosen such that it maximizes the margin, which is the distance between the hyperplane and the closest data points from each class. This approach allows the algorithm to generalize well to new data points.

SVMs are particularly useful in cases where the data is not linearly separable. In such cases, the algorithm can use kernel functions to transform the data into a higher-dimensional space where it is more easily separable. This makes SVMs an effective tool for solving complex classification problems.

One of the advantages of SVMs is that they have a solid theoretical foundation, which makes them popular in a wide variety of machine learning applications. However, they can be sensitive to the choice of kernel function and regularization parameter, and the training process can be computationally expensive for large datasets. To improve the performance of SVMs, researchers have developed various techniques such as cross-validation and grid search to optimize the hyperparameters.

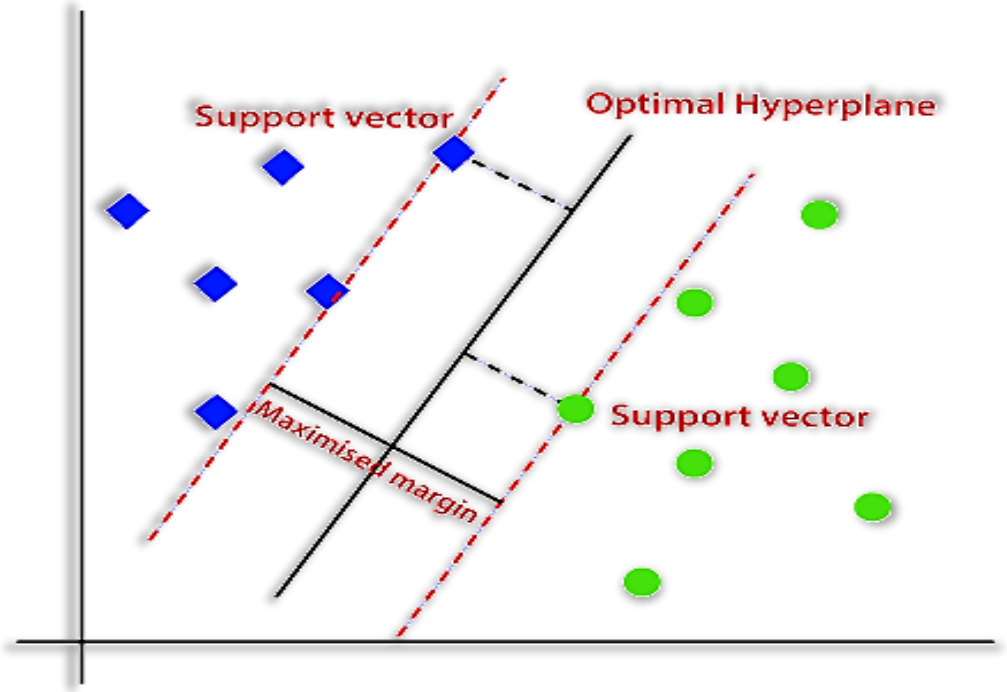


Figure 19

This figure shows how SVMs are modeled. It shows support vector, optimal hyperplane and maximized margin [32]

SVM is applied here. It has given us accuracy of 93%.

Code of this algorithm is given below:

```
In [8]: #Import svm model
        from sklearn import svm

        #Create a svm Classifier
        clf = svm.SVC(kernel='linear') # Linear Kernel

        #Train the model using the training sets
        clf.fit(X_train, y_train)

        #Predict the response for test dataset
        y_pred = clf.predict(X_test)
```

Figure 20

```
In [9]: #Import scikit-learn metrics module for accuracy calculation
        from sklearn import metrics

        # Model Accuracy: how often is the classifier correct?
        print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.9382022471910112
```

Figure 21

4.4 Linear Regression:

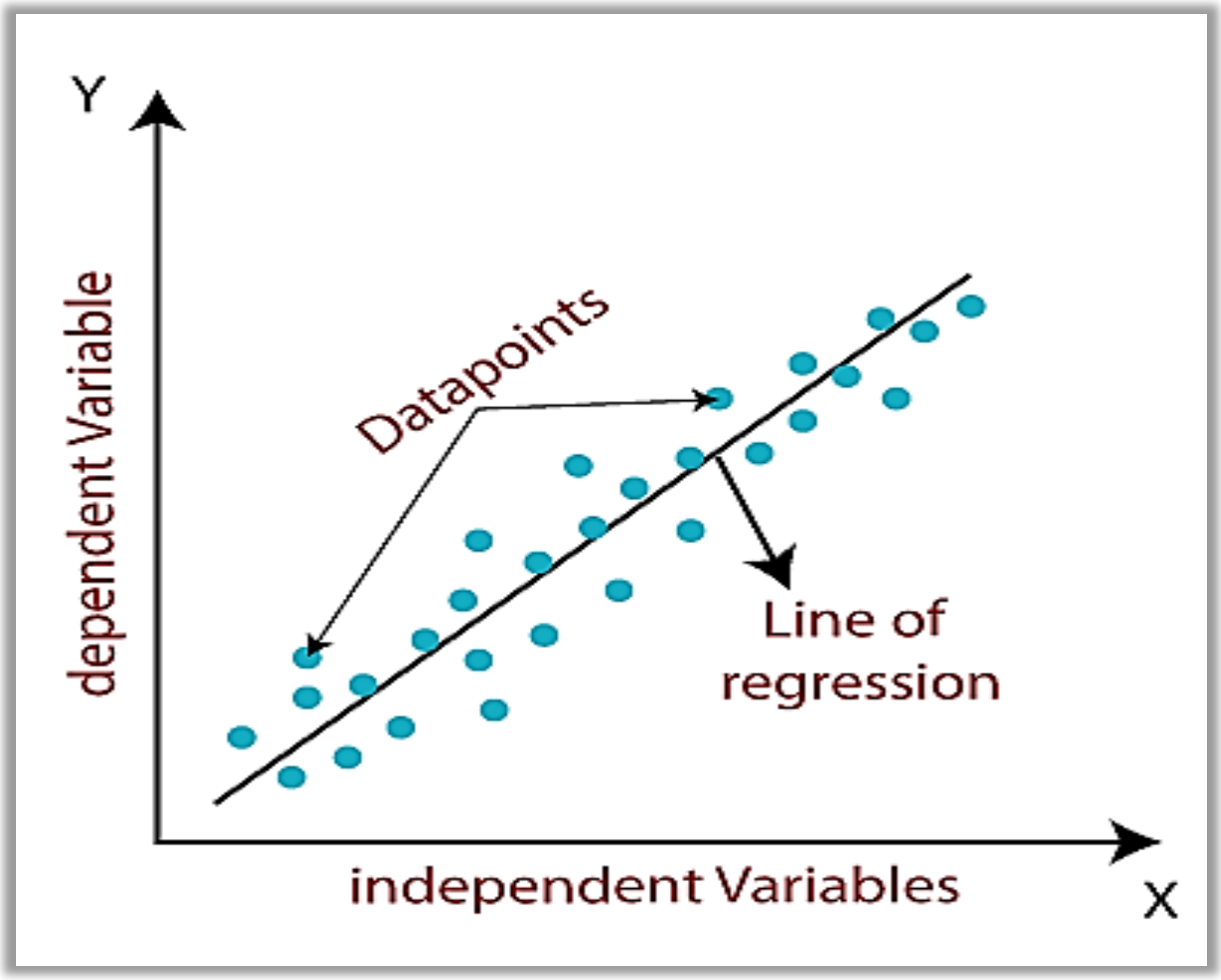
Linear regression is a popular machine learning algorithm that is used to predict continuous outcomes based on one or more input variables. It is a type of supervised learning algorithm that is frequently used in regression analysis.

To fit a linear regression model, the algorithm determines the optimal values of coefficients, denoted by b0, b1, b2, ..., bn, that minimize the sum of squared errors between the predicted and actual values. The model equation is represented by y = b0 + b1x1 + b2x2 + ... + bn*xn, where y is the output variable, and x1, x2, ..., xn are the input variables.

The optimization algorithm used to determine the optimal coefficients is typically a variant of gradient descent. Once the coefficients have been found, the model can be used to make predictions on new data points by plugging in the values of the input variables.

Linear regression is widely used in various fields, including finance, economics, and social sciences, due to its simplicity and effectiveness. However, it has the assumption of a linear relationship between the input and output variables, which may not always hold in real-world

scenarios. To overcome this limitation, various extensions of linear regression, such as polynomial regression and ridge regression, have been developed.



This figure shows how Linear Regression is modeled. It shows dependent variables and independent variables along y-axis and y-axis respectively. Datapoints and line of regression are shown along these axes. [33]

We applied Linear Regression here which gave us 78% accuracy.

Code of it is given below.

```
In [12]: from sklearn import linear_model
         X = df[feature_cols]
         y = df.Label
         lm = linear_model.LinearRegression()
         model = lm.fit(X,y)
         predictions = lm.predict(X)
         print(predictions[0:5])

         [0.06828703 0.02386749 0.0296947  0.059976   0.00465394]

         lm.score(X,y)

In [13]: lm.score(X,y)

Out[13]: 0.7832842425610477
```

Code for Linear Regression

For better understanding, here is the table that shows accuracy of our applied algorithms which are noteworthy, consistently producing results that are close to ground truth.

Sr No.	Algorithm	Accuracy
1.	Decision Tree	100%
2.	Neural Network using TensorFlow	100%
3.	Support Vector Machine (SVM)	93.8%
4.	Linear Regression	78%

CHAPTER 5

DISCUSSION

Smart Buildings are changing the way we approach building design, construction, and management by incorporating advanced technology and systems that offer improved efficiency, reduced costs, and enhanced occupant experience. One of the key benefits of Smart Buildings is their energy efficiency. By using intelligent building automation systems, building owners and managers can optimize energy usage in real-time, leading to cost savings and environmental benefits.

Smart Buildings also prioritize occupant comfort and productivity by incorporating features like lighting and temperature control, advanced ventilation systems, and air quality monitoring. These features create a healthier and more pleasant environment for building occupants, ultimately enhancing their well-being and productivity.

In addition, Smart Buildings offer streamlined maintenance and operations, which result in reduced maintenance costs, enhanced building reliability, and improved property value. The incorporation of access control systems, video surveillance, and emergency response systems also leads to enhanced security and a safer environment for building occupants.

Overall, Smart Buildings provide numerous benefits that revolutionize building design, construction, and management. With continued advancements in technology and innovation, the Smart Buildings industry is poised to continue growing and transforming the built environment in the future.

Smart buildings are gaining popularity for their potential to boost efficiency, cut costs, and provide a better experience for building occupants. Through the integration of advanced technology and systems, these buildings offer cutting-edge features that enhance their functionality, energy-efficiency, and comfort.

One of the key benefits of smart buildings is their energy efficiency. With intelligent building automation systems, building managers and owners can monitor and control energy usage in

real-time, which helps optimize heating, cooling, lighting, and other systems, resulting in energy savings, lower costs, and positive environmental impact.

Apart from energy efficiency, smart buildings also prioritize occupant comfort and productivity. They are designed to include features such as lighting and temperature control to provide a comfortable environment that enhances productivity and well-being. Advanced ventilation systems, air quality monitoring, and other features also promote a healthy and enjoyable environment for occupants.

Figure 22

This diagram shows smart building which is connected to various devices. All of these devices are connected to internet [26]

Smart buildings also provide streamlined maintenance and operations. Building managers can detect and address maintenance issues using IoT sensors and other technology before they develop into larger problems. This results in lower maintenance costs and increased building reliability, which ultimately improves the overall value of the property.

In addition to that, smart buildings have enhanced security features. Building managers can monitor and respond to security threats in real-time using access control systems, video surveillance, and emergency response systems, providing a safer and more secure environment for occupants.

As technology advances, the smart buildings industry is experiencing rapid growth, with new innovations and solutions emerging regularly. The potential benefits of smart buildings are likely to increase with advancements in technology, presenting new and exciting opportunities for building owners, managers, and occupants alike.

In smart buildings, sensors are vital devices that detect changes in the environment and provide real-time data to building managers and automation systems. They are intelligent and small devices that play a crucial role in optimizing various building systems, including lighting, ventilation, heating, and cooling, to enhance energy efficiency and occupant comfort levels.

Temperature sensors are one of the most commonly used sensors in smart buildings. They detect the temperature of the air or water in the building and provide real-time data to the building automation system. This data is used to regulate the heating and cooling systems, ensuring the building remains at a comfortable temperature while minimizing energy consumption.

Figure 23

This image shows various sensors that are used in a smart building. They include heating, cooling, alarm, air quality, light sensors and various others. [27]

When it comes down to monitoring a smart building, the big picture can be segmented as follows:

- Building infrastructure
- Parking
- Smart water
- Elevators and escalators
- Security and emergency
- Access control systems including CCTV
- Safety systems
- Energy management, including heating, ventilation, and air conditioning

- Lighting systems
- Network management
- Waste management

Occupancy sensors are also essential in smart buildings. They detect the presence of individuals in a room or space and can be used to control lighting, heating, and ventilation systems. For example, if a room is empty, the lighting can be automatically turned off, and the heating or cooling system can be adjusted to reduce energy consumption.

Light sensors are another commonly used type of sensor in smart buildings. They detect the amount of natural light entering a space and adjust artificial lighting accordingly. This helps to reduce energy consumption and creates a more comfortable environment for building occupants.

Air quality sensors are becoming increasingly popular in smart buildings. They detect levels of pollutants, carbon dioxide, and humidity in the air and provide real-time data to building managers. This information can be used to adjust the ventilation system to maintain optimal air quality levels, ensuring a healthy and comfortable environment for occupants.

Sensors are an essential component of smart buildings, providing building managers with real-time data to optimize building systems for improved energy efficiency, comfort, and safety. As technology continues to advance, new types of sensors will be developed and integrated into smart buildings to further enhance their capabilities.

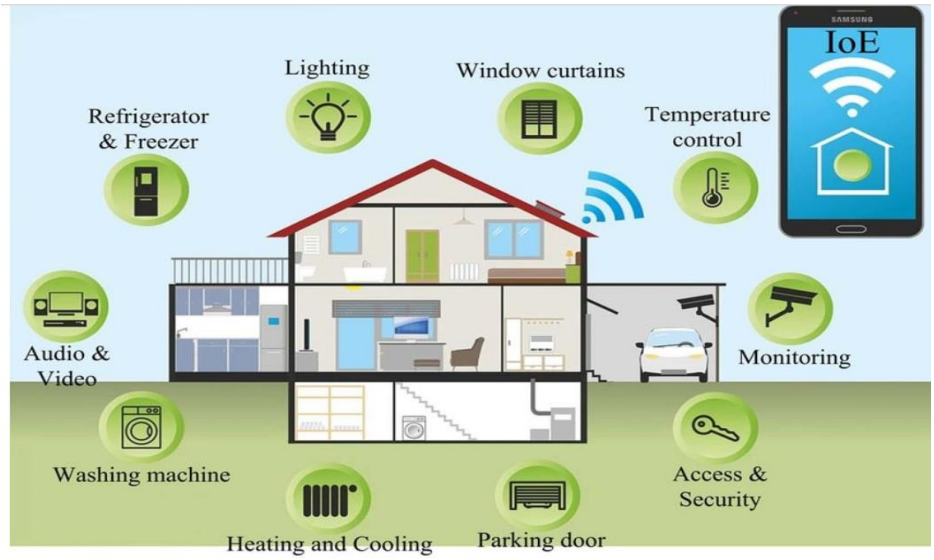


Figure 24

This image shows various sensors that are used in a smart building. They include heating, cooling, alarm, air quality, light sensors and various others. [28]

Smart buildings rely on a variety of sensors to optimize efficiency, security, and comfort. Temperature, occupancy, light, and air quality sensors are common in smart buildings, but there are other types of sensors as well.

Humidity sensors are one such type, which measures the moisture content in the air. By working in tandem with the HVAC system, humidity sensors can adjust humidity levels to create a healthier and more comfortable indoor environment.

Motion sensors are also prevalent in smart buildings, detecting movement to trigger lighting or alert security systems. They can also integrate with the HVAC system to adjust temperature and ventilation based on occupancy.

Water sensors are crucial for detecting leaks or flooding, allowing building managers to take action before significant damage occurs. They can also monitor water usage and identify abnormalities, optimizing water management and reducing costs.

Noise sensors monitor sound levels in different areas of the building, allowing the HVAC system to adjust or activate acoustic dampening features to reduce noise levels in noisy areas.

Advanced sensors are being developed and integrated into smart buildings as well, such as sensors that can detect specific gases or chemicals in industrial settings.

In summary, sensors are an essential part of smart buildings, providing real-time data to enable building managers to optimize systems, reduce costs, and create a more comfortable and secure environment for occupants. As technology continues to advance, new sensors will likely emerge to further improve the efficiency and functionality of smart buildings.

Smart buildings are vulnerable to various cyber-attacks due to their reliance on advanced technology and connected systems. The interconnected nature of these systems means that a single security breach can potentially affect the entire building's operations, including heating, cooling, lighting, and security systems.

The most common types of attacks on smart buildings include Distributed Denial of Service (DDoS), Man-in-the-Middle (MitM), phishing, and malware attacks. DDoS attacks flood a building's network with traffic, causing it to crash and disrupting its operations. MitM attacks intercept communication between connected devices and alter the data exchanged between them. Phishing attacks trick recipients into sharing sensitive information, while malware attacks infect the building's systems and cause significant damage.

To mitigate these risks, building managers need to implement robust security measures such as firewalls, encryption, and access control systems. Regular software updates and patches

should also be applied to ensure that systems are protected against known vulnerabilities. Building managers should also educate employees and occupants about the importance of cybersecurity and best practices for maintaining secure systems. With a comprehensive approach to security, smart buildings can be protected against cyber threats and provide a safe and efficient environment for their occupants.

Smart buildings face various security challenges that can compromise their operations, privacy, and safety. It is crucial for building managers to proactively identify potential risks and implement security measures to protect against attacks.

One of the most significant security risks is the use of unsecured IoT devices. IoT devices like sensors and cameras are connected to the internet and can be easily compromised if not secured correctly. Hackers can exploit these devices to gain unauthorized access to the building's network, steal sensitive data, or even take control of critical systems.

Smart buildings rely on interconnected devices and systems to manage and control building functions, making them vulnerable to DDoS (Distributed Denial of Service) attacks. To prevent and mitigate such attacks in smart buildings, several measures can be taken:

1. Implement robust security measures: Strong firewalls, secure access controls, and encryption technologies can help protect smart buildings against DDoS attacks.
2. Conduct regular security audits: Regular security audits can identify potential vulnerabilities and risks in the building's systems and devices, enabling building managers to fix any security gaps before they are exploited.
3. Monitor network traffic: Regular monitoring of network traffic can help detect unusual activity that may indicate a DDoS attack, allowing building managers to take immediate action to prevent or mitigate the attack.

4. Deploy DDoS protection technologies: Intrusion detection and prevention systems can help identify and block malicious traffic before it reaches the building's systems.
5. Create a response plan: A response plan should be in place in case of a DDoS attack, including clear procedures for identifying, reporting, and responding to an attack, as well as a communication plan to keep all stakeholders informed.

Overall, a combination of strong security measures, regular monitoring, and a comprehensive response plan is necessary to prevent and mitigate DDoS attacks in smart buildings. Cybersecurity should be a top priority for building managers to ensure the safety and functionality of their smart building systems.

The lack of standardization in smart building technologies is another significant challenge. Different vendors may use different protocols and security measures, making it difficult to integrate and manage these systems cohesively. This can create vulnerabilities that hackers can exploit to gain unauthorized access to the building's systems.

Physical security is also a critical consideration for smart buildings. Building managers must ensure that physical access to critical systems such as HVAC, lighting, and security is restricted only to authorized personnel. Unauthorized access can compromise these systems, leading to downtime or even physical harm to occupants.

To address these challenges, building managers must implement a comprehensive security strategy that includes:

1. Regular security assessments to identify potential vulnerabilities and risks.
2. Robust access control measures, such as biometric authentication and authorization protocols.

3. Network security solutions like firewalls, intrusion detection and prevention systems, and encryption to protect the building's network against cyber attacks.
4. Educating employees and occupants about cybersecurity best practices and the risks associated with unsecured IoT devices.
5. Regularly updating and patching software and firmware to ensure systems are protected against known vulnerabilities.
6. Working with vendors to ensure that all IoT devices and systems are secure and compatible with the building's network and security protocols.

In conclusion, smart buildings face numerous security challenges that require proactive measures to mitigate risks. Building managers must prioritize security and implement a comprehensive strategy to ensure the safety, privacy, and efficiency of their buildings.

Simulation of smart building sensors is essential for designing and testing smart building systems. It allows engineers and managers to detect potential issues with sensors and fine-tune the systems before actual deployment.

Smart building sensors can monitor different environmental factors, such as temperature, humidity, air quality, occupancy, and lighting, to optimize building operations, enhance energy efficiency, and improve occupant safety and comfort.

To simulate smart building sensors, a virtual environment that emulates the physical environment of the building is created, including the building's layout, HVAC system, lighting, and other relevant components. Simulated sensors are then integrated into the virtual environment to produce data that reflects real-world sensors' behaviour.

Simulated sensor data is analysed and compared to expected results to identify discrepancies or sensor issues, which helps to fine-tune the sensors and improve system performance, accuracy, and efficiency.

Simulating smart building sensors also allows testing of different scenarios that may not be safe or feasible to test in the actual building. For instance, emergency scenarios like fire or natural disasters can be simulated to evaluate the sensors and building systems' response.

Figure 25

Overall diagram showing snippet of smart building [29]

Digitalization in the building sector pertains to a collection of digital technologies that enable the acquisition of valuable data about a building and its communication to relevant parties. While innovation in the building sector has typically cantered around materials and construction methods, digitalization introduces new possibilities in building management.

In summary, simulating smart building sensors is critical for effective smart building system design and deployment. By optimizing sensor performance, building managers can ensure their buildings' safety, comfort, and efficiency.

Smart buildings can reduce their carbon footprint and energy consumption significantly by utilizing advanced technologies such as IoT sensors, building automation systems, and data analytics. For instance, smart HVAC systems can monitor real-time occupancy and weather data to adjust temperature and airflow, thus decreasing energy waste and enhancing indoor air quality. Smart lighting systems can also optimize lighting levels based on natural light and occupancy, reducing energy usage and enhancing the comfort and productivity of occupants.

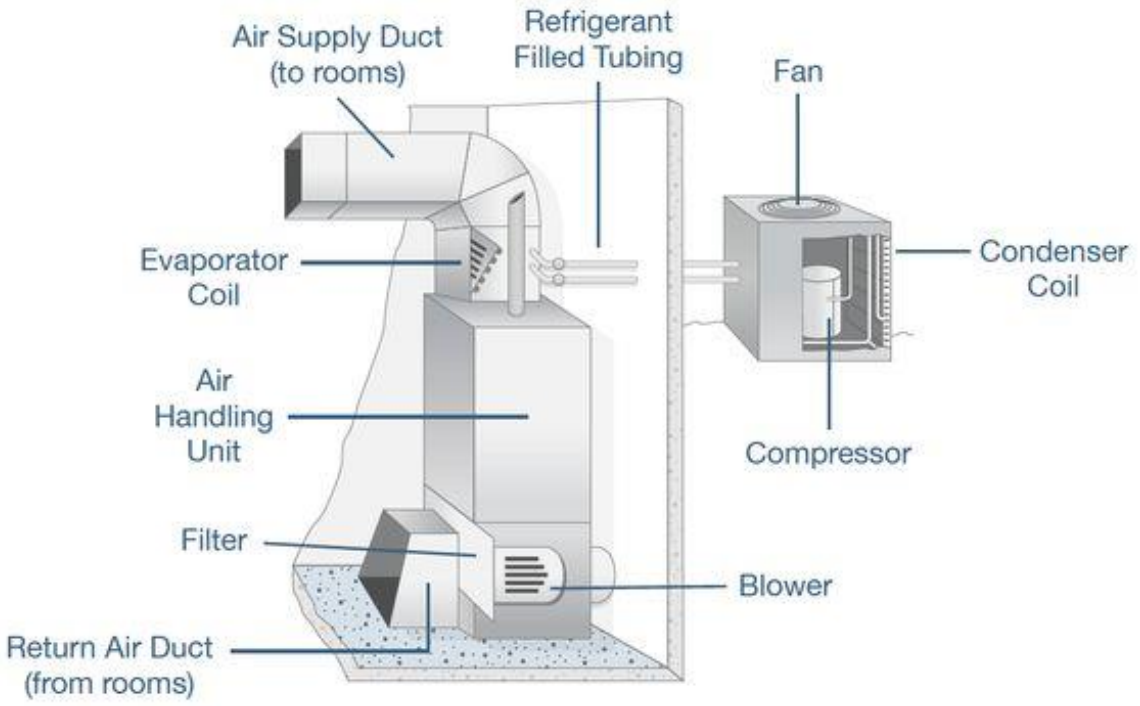


Figure 26

Air conditioning system in HVAC [28]

HVAC systems are essential elements of contemporary constructions designed to maintain optimal indoor environmental conditions, including temperature, humidity, and air quality. These systems comprise various components, including air filters, heat pumps, air handlers, ductwork, and thermostats, all working in conjunction to regulate the indoor environment effectively and efficiently. As environmental concerns gain prominence, the significance of HVAC systems has increased as they play an important role in achieving energy efficiency and sustainability in buildings.

The technical and scientific aspects of HVAC systems are intricate and require specialized expertise to design and optimize. The selection of appropriate components, such as efficient heat pumps and properly sized ductwork, is crucial in ensuring the optimal performance of HVAC systems. Additionally, the appropriate installation and maintenance of these systems are critical in maintaining their energy efficiency and reliability over time.

In the context of sustainability, HVAC systems can significantly influence a building's energy consumption and carbon footprint. The proper design and operation of HVAC systems can lead to substantial energy savings, contributing to a more sustainable future.

To summarize, HVAC systems are crucial components of modern buildings that necessitate technical proficiency for design and optimization. Proper installation and maintenance of HVAC systems are necessary to sustain their energy efficiency and reliability over time, leading to significant energy savings and a more sustainable future.

Here is some more information about HVAC systems:

- HVAC systems are designed not only for heating and cooling but also for regulating humidity levels. This feature is crucial for both health and comfort, as excessive humidity can cause mould growth and health issues, while low humidity can lead to dry skin, throat irritation, and static electricity build-up.

- HVAC systems have undergone significant advancements in energy efficiency. Thanks to technological progress such as variable-speed compressors, programmable thermostats, and smart controls, contemporary systems consume less energy and produce fewer emissions compared to earlier models.
- In an effort to reduce their carbon footprint, HVAC systems can incorporate renewable energy sources like solar panels and geothermal systems.
- Ductless HVAC systems, also known as mini-split systems, are gaining popularity in small spaces like homes and buildings. These systems do not require ductwork and can be more energy-efficient than conventional HVAC systems.
- HVAC systems also have the ability to contribute to indoor air quality by filtering out pollutants and allergens. Some systems use specialized filters, such as HEPA filters, to capture smaller particles and enhance air quality.

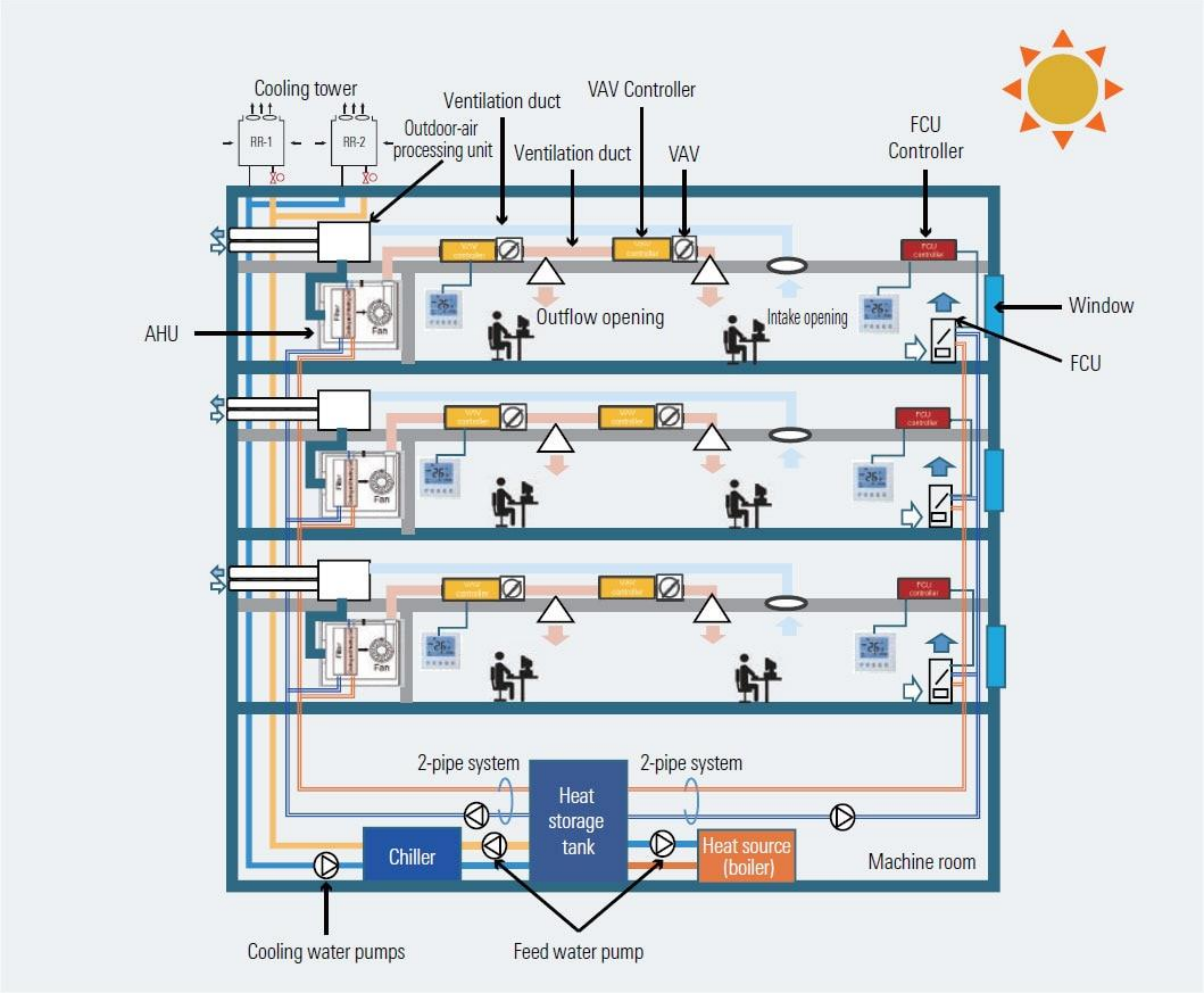


Figure 27

Cooling and Heating system in a HVAC system. It shows detailed picture of the placement of each object in this system. [28]

Smart buildings can also incorporate renewable energy sources, such as solar panels and wind turbines, to generate clean energy and minimize reliance on fossil fuels. Additionally, energy storage systems like batteries can be integrated to provide backup power during grid outages, further reducing energy costs.

To minimize waste and greenhouse gas emissions, smart buildings can deploy waste management solutions that incorporate IoT sensors and data analytics to optimize waste

collection and recycling processes. This can enhance the cleanliness and hygiene of the building while reducing waste.

This thesis proposes a machine learning (ML) based approach to detect DDoS attacks in smart buildings. The proposed solution employs various ML algorithms, including SVM, decision trees, and linear regression. These models are trained to analyse network traffic data collected from smart building devices and detect and classify network traffic patterns that indicate DDoS attacks.

Our proposed algorithm gave us really promising results. Decision Tree gave us 100% accuracy, SVM gave us 93% accuracy where linear regression gave us 78% accuracy

This section discusses the challenges that smart buildings face in terms of security and emphasizes the need for building managers to implement proactive measures to mitigate risks. Simulating smart building sensors is critical for effective smart building system design and deployment.

As we are using dataset which is generated by the simulator, we think that there might be some limitations. As we know that simulated data is frequently used in cybersecurity to train machine learning models, including for detecting DDoS attacks in smart buildings. However, while simulated data can be useful in some contexts, it also has several limitations that can impact the effectiveness of the model in real-world scenarios.

One limitation of simulated data is that it may not accurately represent the diversity of real-world data. The data used to generate simulations may not fully capture the complexity and unpredictability of actual network traffic, leading to oversimplified models that are not effective in identifying new types of attacks.

Another limitation of simulated data is that it may not accurately reflect the distribution of traffic in real-world environments. Simulated data may not capture the variability in network traffic that can result from factors such as changes in user behaviour or seasonal trends. As a result, models that are poorly calibrated to the real-world environment may generate high false positive or false negative rates.

Finally, the use of simulated data may introduce biases into the model that can affect its performance in detecting real-world attacks. For example, the simulated data may be biased towards certain types of attacks or network configurations, resulting in a less effective model in detecting attacks in other contexts.

Smart building sensors can monitor different environmental factors to optimize building operations, enhance energy efficiency, and improve occupant safety and comfort. HVAC systems are crucial components of modern buildings that require technical proficiency for design and optimization. Proper installation and maintenance of HVAC systems are necessary to sustain their energy efficiency and reliability over time, leading to significant energy savings and a more sustainable future. Smart buildings can incorporate renewable energy sources, waste management solutions, and energy storage systems to generate clean energy and minimize waste. Additionally, this section proposes a machine learning-based approach to detect DDoS attacks in smart buildings, which yielded promising results. Limitations of using dataset generated by simulator is also discussed.

CHAPTER 6

CONCLUSION

Smart buildings, enabled by the widespread use of Internet of Things (IoT) devices, are becoming increasingly prevalent. However, this rise in IoT adoption also brings new security challenges, with smart buildings more vulnerable to cyber-attacks, including distributed denial-of-service (DDoS) attacks. DDoS attacks can cause significant damage to the building's network infrastructure, leading to financial losses and downtime.

This thesis proposes a machine learning (ML) based approach to detect DDoS attacks in smart buildings. The proposed solution employs various ML algorithms, including SVM, decision trees, Neural Network using TensorFlow and linear regression. These models are trained to analyse network traffic data collected from smart building devices and detect and classify network traffic patterns that indicate DDoS attacks.

Our proposed algorithm gave us really promising results. Decision Tree gave us 100% accuracy, Neural Network using TensorFlow gave us 100% accuracy, SVM gave us 93% accuracy where linear regression gave us 78% accuracy

There are some limitations of this study too. Although simulated data is frequently used in cybersecurity to train machine learning models for detecting DDoS attacks in smart buildings, its effectiveness in real-world scenarios can be impacted by several limitations. One such limitation is that the simulated data may not accurately represent the diversity and complexity of real-world data, which can lead to oversimplified models that are not effective in identifying new types of attacks. Additionally, simulated data may not reflect the distribution of traffic in real-world environments due to factors such as changes in user behavior or seasonal trends, resulting in high false positive or false negative rates. Furthermore, the use of simulated data may introduce biases into the model, leading to a less effective model in detecting attacks in other contexts that the simulated data may not account

for. The study acknowledges these limitations and has attempted to carefully craft its methodology to address these concerns.

Chapter 7

REFERENCES

[1] Elnour, M., Meskin, N., Khan, K., & Jain, R. (2021). Application of data-driven attack detection framework for secure operation in smart buildings. *Sustainable Cities and Society*, *69*, 102816. https://doi.org/10.1016/j.scs.2021.102816

[2] Hachem, J. E., Chiprianov, V., Babar, M. A., Khalil, T. A., & Aniorte, P. (2020). Modeling, analyzing and predicting security cascading attacks in smart buildings systems-of-systems. *Journal of Systems and Software*, *162*, 110484. https://doi.org/10.1016/j.jss.2019.110484

[3] Alanne, K., & Sierla, S. (2022). An overview of machine learning applications for smart buildings. *Sustainable Cities and Society*, *76*, 103445. https://doi.org/10.1016/j.scs.2021.103445

[4] Alhaidari, F. A., & AL-Dahasi, E. M. (2019, April 1). New Approach to Determine DDoS Attack Patterns on SCADA System Using Machine Learning. IEEE Xplore. https://doi.org/10.1109/ICCISci.2019.8716432

[5] Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms, 2021,Vinícius de Miranda Rios, Pedro R.M. Inácio, Damien Magoni, Mário M. Freire,10.1016/j.comnet.2020.107792, Computer Networks

[6] Gaurav, A., Gupta, B. B., Hsu, C.-H., Yamaguchi, S., & Chui, K. T. (2021, January 1). *Fog Layer-based DDoS attack Detection Approach for Internet-of-Things (IoTs) devices*. IEEE Xplore. https://doi.org/10.1109/ICCE50685.2021.9427648

[7] Syed, N. F., Baig, Z., Ibrahim, A., & Valli, C. (2020). Denial of service attack detection through machine learning for the IoT. *Journal of Information and Telecommunication*, *4*(4), 482–503. https://doi.org/10.1080/24751839.2020.1767484

[8] Gaber, T., El-Ghamry, A., & Hassanein, A. E. (2022). Injection attack detection using machine learning for smart IoT applications. Physical Communication, 101685. https://doi.org/10.1016/j.phycom.2022.101685

[9] Ali, T. E., Chong, Y.-W., & Manickam, S. (2023). Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. *Applied Sciences*, *13*(5), 3183. https://doi.org/10.3390/app13053183

[10] Kumari, K., & Mrunalini, M. (2022). Detecting Denial of Service attacks using machine learning algorithms. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00616-0

[11] Musumeci, F., Fidanci, A. C., Paolucci, F., Cugini, F., & Tornatore, M. (2021). Machine-Learning-Enabled DDoS Attacks Detection in P4 Programmable Networks. *Journal of Network and Systems Management*, *30*(1). https://doi.org/10.1007/s10922-021-09633-5

[12] Singh Saini, P., & Behal, S. (2020). Detection of DDoS Attacks using Machine Learning Algorithms [Review of *Detection of DDoS Attacks using Machine Learning Algorithms*]. *IEEE International Conference*. https://doi.org/10.23919/INDIACom49435.2020.9083716

[13] DDoS Attack Detection Using Hybrid Machine Learning Based IDS Models. (2022). *Journal of Scientific & Industrial Research*, *81*(03). https://doi.org/10.56042/jsir.v81i03.58451

[14] He, Z., Zhang, T., & Lee, R. (n.d.). *Machine Learning Based DDoS Attack Detection From Source Side in Cloud*. Retrieved April 8, 2023, from http://palms.princeton.edu/system/files/Machine_Learning_Based_DDoS_Attack_Detection From_Source_Side_in_Cloud_camera_ready.pdf

[15] Gupta, S., Grover, D., Ali AlZubi, A., Sachdeva, N., Waqar Baig, M., & Singla, J. (2022). Machine Learning with Dimensionality Reduction for DDoS Attack Detection. Computers, Materials & Continua, 72(2), 2665–2682. https://doi.org/10.32604/cmc.2022.025048

[16] Yu, X., Yu, W., Li, S., Yang, X., Chen, Y., & Lu, H. (2021). WEB DDoS Attack Detection Method Based on Semi supervised Learning. Security and Communication Networks, 2021, 1–10. https://doi.org/10.1155/2021/9534016

[17] Karthika, P., & Arockiasamy, K. (2023). Simulation of SDN in mininet and detection of DDoS attack using machine learning. Bulletin of Electrical Engineering and Informatics, 12(3), 1797–1805. https://doi.org/10.11591/eei.v12i3.5232

[18] Rahman, M. A. (2020). Detection of Distributed Denial of Service Attacks based on Machine Learning Algorithms. International Journal of Smart Home, 14(2), 15–24. https://doi.org/10.21742/ijsh.2020.14.2.02

[19] Naveen Bindra, & Manu Sood. (2019). Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset. Automatic Control and Computer Sciences, 53(5), 419–428. https://doi.org/10.3103/s0146411619050043

[20] Gupta. (2018). Distributed Denial of Service Attack Detection Using a Machine Learning Approach. https://doi.org/10.11575/PRISM/32797

[21] Alkasassbeh, M., Al-Naymat, G., B.A, A., & Almseidin, M. (2016). Detecting Distributed Denial of Service Attacks Using Data Mining Techniques. *International Journal of Advanced Computer Science and Applications, 7*(1). https://doi.org/10.14569/ijacsa.2016.070159

[22] Lin, Y.-S., & Lee, C.-F. (2023). Ransomware Detection and Prevention through

Strategically Hidden Decoy File. International Journal of

Network Security, 25(2), 212–220. https://doi.org/10.6633/IJNS.202303

[23] Baich, M., Hamim, T., Sael, N., & Chemlal, Y. (2022).

Machine Learning for IoT based networks intrusion detection: a comparative study.

Procedia Computer Science, 215, 742–751. https://doi.org/10.1016/j.procs.2022.12.076

[24] S. Priya, M. Sivaram, D. Yuvaraj, & A. Jayanthiladevi. (2020).

Machine Learning based DDOS Detection. 2020 International Conference on Emerging

Smart Computing and Informatics (ESCI).

https://www.semanticscholar.org/paper/Machine-Learning-based-DDOS-Detection-Priya-Sivaram/bb9ade7b3c6fada32e964c883b7f43498ba48b00

[25] Esmaeili, M., Goki, S. H., Masjidi, B. H. K., Sameh, M., Gharagozlou, H., & Mohammed, A. S. (2022). ML-DDoSnet: IoT Intrusion Detection Based on Denial-of-Service Attacks Using Machine Learning Methods and NSL-KDD. Wireless Communications and Mobile Computing, 2022, 1–16. https://doi.org/10.1155/2022/8481452

[26] *Smart Building*. (n.d.). Www.paessler.com. Retrieved April 23, 2023, from https://www.paessler.com/fr/iot/smart-building

[27] Energisme-tCq7YeY72. (2021, July 22). *Smart building: 9 good reasons to optimize your building energy management - Energisme*. https://energisme.com/en/smart-building-9-good-reasons-to-optimize-your-building-energy-management/

[28] *HVAC Systems | Renesas*. (2020). Renesas.com. https://www.renesas.com/tw/en/application/industrial/building-home-automation/hvac-systems

[29] IBM. (2022). *What is a Decision Tree | IBM*. Www.ibm.com. https://www.ibm.com/topics/decision-trees

[30] *Multi-Layer Perceptron Learning in Tensorflow*. (2021, November 3). GeeksforGeeks. https://www.geeksforgeeks.org/multi-layer-perceptron-learning-in-tensorflow/

[31] *Building Neural Networks in TensorFlow*. (n.d.). Ryan P. Marchildon. Retrieved April 27, 2023, from https://rpmarchildon.com/ai-cnn-digits

[32] javatpoint. (n.d.). *Support Vector Machine (SVM) Algorithm - Javatpoint*. Www.javatpoint.com. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[33] *Linear Regression in Machine learning - Javatpoint*. (n.d.). Www.javatpoint.com. https://www.javatpoint.com/linear-regression-in-machine-learning

[34] hambali, moshood & yakub, saheed & oladele, tinuke & gbolagade, morufat. (2019). adaboost ensemble algorithms for breast cancer classification. https://www.researchgate.net/publication/338528758_ADABOOST_ENSEMBLE_ALGORITHMS_FOR_BREAST_CANCER_CLASSIFICATION

[35] Gate, R. (2021). *130+ million publications organized by topic on ResearchGate*. ResearchGate; ResearchGate. https://www.researchgate.net/publication