International Journal of Computer Applications Technology and Research
Volume 13–Issue 04, 01 – 06, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1304.1001

Detection of on-Target Payload Object Drop and Automatic Alert Generator

Sandeep Kumar
Department of CSE
ABES Institute of Technology
Ghaziabad, India

Kshitij Parashar
Department of CSE(AI)
ABES Institute of Technology
Ghaziabad, India

Shruti Sharma
Department of CSE(AI)
ABES Institute of Technology
Ghaziabad, India

Shalini Yadav
Department of CSE(AI)
ABES Institute of Technology
Ghaziabad, India

Vineet Kumar Singh
Department of CSE(AI)
ABES Institute of Technology
Ghaziabad, India

Abstract: In today's dynamic industrial landscape, the need for advanced systems capable of swiftly detecting and responding to on-target or off-target payload object drops in real-time has become increasingly critical. This research paper intro-duces a groundbreaking initiative aimed at revolutionizing safety protocols, optimizing cargo handling procedures, and enhancing operational efficiency through the development of an innovative solution: the "Detection of On-Target Payload Object Drop and Automatic Alert Generator." Acknowledging the pressing demand for accurate, automated, and immediate responses to on-target payload object drops across various sectors including logistics, manufacturing, and security, this research addresses the inherent challenges posed by the lack of a reliable and real-time detection system. Operational disruptions, cargo damage, worker safety concerns, and inefficiencies resulting from accidental drops of payload objects have underscored the urgent need for proactive measures. The proposed system leverages cutting-edge technologies such as the YOLO-NAS deep learning model and the NVidia DGX A100 GPU server to deliver unparalleled performance and precision. By harnessing the power of artificial intelligence and high-performance computing, this intelligent system is designed to accurately detect payload object drops and generate automatic alerts in real-time, tailored to the specific requirements of clients or applications. By integrating seamlessly into existing workflows, this innovative solution promises to not only mitigate the risks associated with payload object drops but also streamline operations, minimize downtime, and uphold the highest standards of safety and efficiency. Through a comprehensive analysis of industry-specific challenges and practical implementation strategies, this research paper offers valuable insights into the transformative potential of advanced detection systems in safeguarding valuable cargo, protecting personnel, and optimizing operational performance.

Keywords: Payload Object Detection; YOLO-NAS; Nvidia DGX A100; Real-Time Alert System; Security; Safety; Automation.

1. INTRODUCTION

In today's rapidly evolving industrial landscape, the effective management and safe-guarding of cargo during transportation are of paramount importance. The safe delivery of payload objects is not only crucial for maintaining operational efficiency but also for ensuring the safety of workers and minimizing financial losses associated with cargo damage. However, despite advancements in logistics and manufacturing technologies, the occurrence of on-target payload object drops remains a persistent challenge. The lack of a reliable and real-time detection system for on-target payload object drops poses significant risks across various sectors, including logistics, manufacturing, and security. Operational disruptions, cargo damage, worker safety concerns, and inefficiencies resulting from accidental drops underscore the urgent need for pro-active measures to address this issue. Traditional methods of monitoring and respond-ing to such incidents are often manual, time-consuming, and prone to inaccuracies, highlighting the necessity for innovative solutions that can provide accurate, automat-ed, and immediate responses.

To address these challenges, this research paper introduces a groundbreaking initiative: the "Detection of On-Target Payload Object Drop and Automatic Alert Genera-tor." This advanced system is designed to revolutionize safety protocols, optimize cargo handling procedures, and enhance operational efficiency by accurately detecting on-target payload object drops in real-time and generating automatic alerts and notifications for timely response to any inaccuracies or violations.

By leveraging state-of-the-art technologies such as the YOLO-NAS deep learning model and the NVidia DGX A100 GPU Acceleration Server, this intelligent solution promises superior performance and precision. By integrating seamlessly into existing workflows, it aims to mitigate the risks associated with on-target payload object drops while streamlining operations, minimizing downtime, and upholding the highest standards of safety and efficiency. Through a comprehensive analysis of industry-specific challenges and practical implementation strategies, this research paper aims to shed light on the transformative potential of advanced detection systems in safeguarding valuable cargo, protecting personnel, and optimizing operational performance in the logistics, manufacturing, and security sectors.

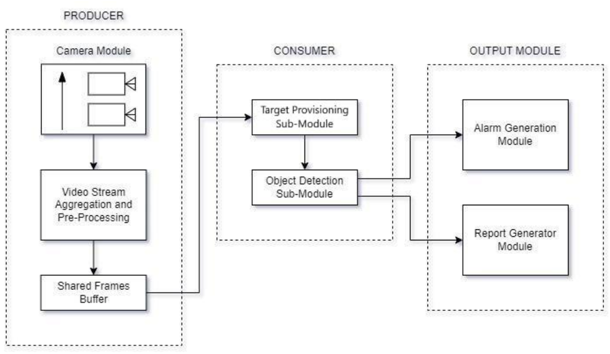


Figure 1. System Diagram

2. LITERATURE SURVEY

Various research and advancements have been carried out in the near past to solve existing problems but very few researches have been on the need to increase the port-ability of the system.

In 2022, Vadym Slyusar et al. proposed an approach using the VisDroneYOLOv5x object detection model specially built for the applications involving the unmanned aerial vehicles. The VisDrone dataset is a collection of images taken from several drones, quadcopters, and many other aerial vehicles that treat the special purpose of training object detection models to detect objects from higher altitudes. The proposed model has less mAP as compared to the newer and upgraded versions. Also, the pro-posed methodology only focused on improving the object detection model rather than building a ready to deploy system. [1]

In 2021, SivaNagi Reddy Kalli et al. proposed a motion object detection using adaptive background modelling mechanism for the video surveillance systems. The approach focused on BMBIFFCM algorithm. In this, the non-stationary pixels are separated from stationary pixels through the Background Subtraction. Afterward, the Biased Illumination Field Fuzzy C-means approach had accomplished to improve the segmentation accuracy through clustering under noise and varying illumination conditions. The performance of the proposed algorithm was compared with conventional methods in terms of accuracy, precision, recall, and F- measure. The significant limitation was the performance of object detection being slower and the features getting reduced in the images. [2]

In 2021, M. Luthfi Hakim et al. proposed an autonomous quadcopter system equipped with an image object detection method to assist COVID-19 patients. The implementation of efficient object detection algorithms enables the quadcopter to accurately identify and deliver assistance payloads to patients in need. The proposed system was only limited to detect QR codes and detect the targets on basis of QR codes. [3]

James Sewell et al. present a deep learning-based approach for real-time object detection in autonomous vehicles. By employing deep neural networks, the proposed system achieves high accuracy in detecting objects in complex traffic environments, con-tributing to the safety and reliability of autonomous driving systems. The use of Hough Circle Transform algorithm in this approach for detecting circles in imperfect images limits the system for detecting only circular target sites and not the objects. Also, the alert generation is not a feature of this system. [4]

Oyku Sahin et al. proposed their implementation of the VisDroneYOLOv3 object detection model. The model used the VisDrone dataset that is limited to drone images only and the use of YOLOv3 leaves the performance metrics of this system a little bit behind as compared to the modern state-of-the-art object detection algorithms and models. [5]

Sudan Jha et al. proposed a tracking system for video surveillance system. Instead of a CNN, a single neural network is applied to the full image. It then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. They therefore proposed their own approach called N-YOLO. It divides the image into fixed sized images using correlation-based tracking algorithm, instead of resizing image step in YOLO algorithm. [6]

Mate Kristo et al. proposed “Thermal object detection in difficult weather conditions using YOLO” which is an advancement and improvisation required in many object detection systems. The system eliminates the potential barrier of bad weather conditions like fogging, rain, and less visibility. Their methodology involves the use of thermal imaging in combination with the YOLO. The only limitation of this approach is that it is not able to determine the objects based on color. [7]

In 2020, Bo Yang et al. proposed an efficient object detection approach leveraging edge computing in IoT environments. By offloading computational tasks to edge de-vices, the proposed system minimizes latency and improves efficiency in object detection applications, making it suitable for real-time IoT deployments. The increase in number of computational devices increases the complexity of the system and hence reducing the portability of the system. [8]

In summary, recent research efforts have focused on enhancing object detection accuracy and efficiency through innovative approaches, including deep learning-based methods, fusion strategies, compressed sensing techniques, and edge computing solutions. These advancements contribute to the development of reliable and efficient systems for various applications, including autonomous navigation, surveillance, and medical assistance. The dedicated system for deployment on target sites for detecting the correct on-target drop of the payload has not been proposed till now, making this research a new introduction to the field of computer vision and alert systems in logistics, defense, and industrial applications.

3. METHODOLOGY

3.1 Custom Target Provisioning Module

The custom Target Provisioning System plays a pivotal role within the "Detection of On-Target Payload Object Drop and Automatic Alert Generator" framework, facilitating precise localization and isolation of regions of interest (ROI) within incoming video frames. This system is designed to dynamically provision target areas based on user-defined parameters, including height, width, and coordinates (x, y), thereby masking irrelevant portions of the video feed and focusing exclusively on the specified ROI.

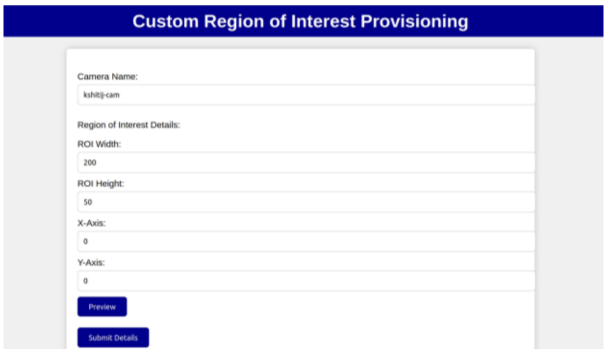
3.1.1 System Architecture

The Target Provisioning System consists of several key components, including:

Parameter Input Module: This module allows users to input specific parameters such as height, width, and coordinates (x, y) defining the ROI within the video frame. The user can preview the ROI on video stream before submitting the coordinates and pro-visioning the target.

ROI Masking Algorithm: Leveraging the user-defined parameters, this algorithm dynamically generates a binary mask wherein the ROI is delineated while the rest of the frame is masked with a black overlay.

Frame Processing Pipeline: Integrating the ROI mask with incoming video frames, this pipeline ensures that only the designated ROI is retained for further analysis, enhancing computational efficiency, and reducing unnecessary processing overhead.


Custom Region of Interest Provisioning

Camera Name:
kshitij-cam

Region of Interest Details:

ROI Width:
200

ROI Height:
50

X-Axis:
0

Y-Axis:
0

Figure 2. Target Provisioning System Interface

3.1.2 *Functionality*

Upon receiving user-defined parameters, the Target Provisioning System dynamically generates a binary mask representing the specified ROI. This mask is then overlaid onto incoming video frames, effectively masking out extraneous regions, and isolating the ROI for subsequent analysis. The provided ROI is verified using the "preview" functionality. But once the parameters are submitted finally, the actual coordinates are updated in the source file where the coordinates of the video stream are declared.

3.1.3 *Implementation*

The Target Provisioning System is implemented using a blend of OpenCV, a robust computer vision library, and custom algorithms tailored for real-time ROI masking. This implementation begins with the establishment of a streamlined video frame processing pipeline, leveraging OpenCV's functions for efficient frame loading and preparation. User-defined parameters such as ROI height, width, and coordinates (x, y) are extracted to dynamically generate a binary mask delineating the specified ROI. Custom algorithms handle this task, ensuring precise spatial manipulation and pixel-level operations to create the mask. In real-time, the generated ROI mask is applied to in-coming video frames, effectively isolating the designated area for further analysis while masking out irrelevant regions. Careful optimizations are integrated to maintain computational efficiency without compromising accuracy, striking a balance between speed and precision. This implementation results in a responsive and adaptable Target Provisioning System, capable of dynamically provisioning and masking ROIs within video frames in real-time. By seamlessly integrating OpenCV and custom algorithms, the system lays the groundwork for accurate detection of on-target payload object drops within the proposed framework. The complete process of frame masking is visualized as follows:

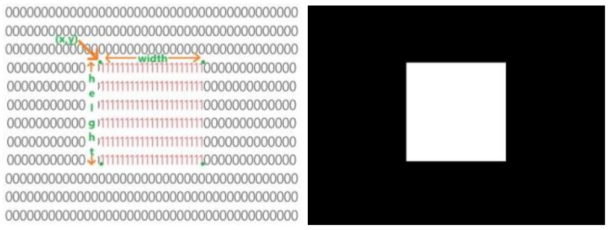


Figure 3. Binary Mask (Numerical and RGB)

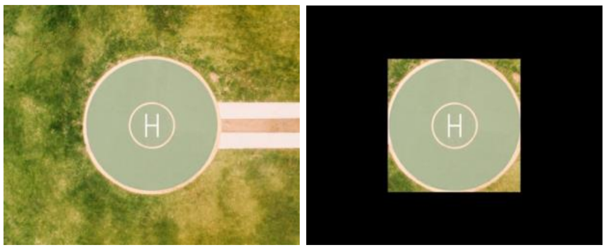


Figure 4. Target Frame (Non-Masked and Masked)

3.1.4 *Integration with Object Detection Module*

The output of the Target Provisioning System, comprising masked video frames with isolated ROI, seamlessly integrates with the Object Detection Module powered by the YOLO-NAS model. This integration ensures that only relevant regions containing potential payload objects are analyzed, optimizing system efficiency, and enhancing the overall detection accuracy.

By providing a customizable and adaptive solution for isolating regions of interest within video frames, the Target Provisioning System enhances the capabilities of the overall system in accurately detecting on-target payload object drops and generating automatic alerts in real-time.

3.2 Object Detection Module

The Object Detection Module implemented in the "Detection of On-Target Payload Object Drop and Automatic Alert Generator" system utilizes the YOLO-NAS (You Only Look Once - Neural Architecture Search) deep learning model. YOLO-NAS is renowned for its superior real-time object detection capabilities and high performance, making it an ideal choice for our system's requirements.

3.2.1 *YOLO-NAS Architecture*

The YOLO-NAS architecture is characterized by its efficient design, which allows for rapid and accurate detection of objects in real-time. Unlike traditional object detection models that rely on complex multi-stage pipelines, YOLO-NAS adopts a single-stage approach, enabling faster inference without compromising accuracy.

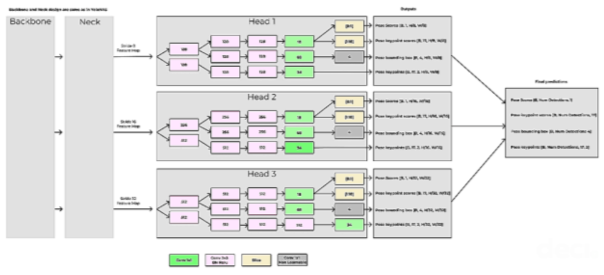


Figure 5. YOLO-NAS Architecture – Head Design

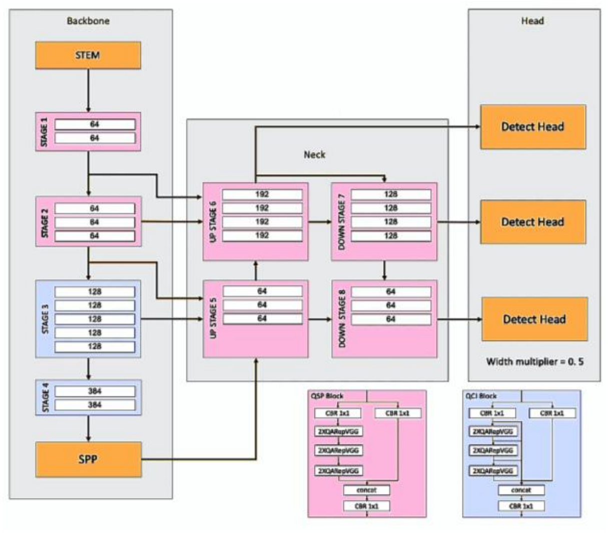


Figure 6. YOLO-NAS Architecture – Backbone and Neck Design

3.2.2 Model Training

The YOLO-NAS model is trained using a large dataset of annotated images containing various payload objects commonly encountered in logistics, manufacturing, and security scenarios. Through an iterative process, the model learns to accurately detect and localize these objects within images, taking account of factors such as size, orientation, and occlusion.

3.2.3 Real-Time Inference

Once trained, the YOLO-NAS model is deployed within the Object Detection Module of our system, where it performs real-time inference on incoming video streams or images. By analyzing each frame or image in its entirety, YOLO-NAS efficiently identifies and classifies payload objects present within the scene, enabling prompt detection of on-target drops.

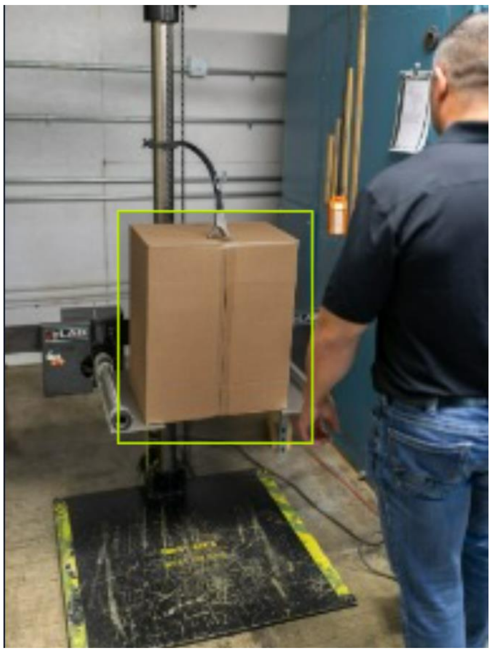


Figure 7. Live Inference using the Trained Model

3.2.4 Performance Optimization

To ensure optimal performance, the YOLO-NAS model is optimized for training as well as deployment on GPU hardware platforms such as the NVidia DGX A100. Leveraging the computational power of the DGX A100, the model achieves unprecedented speed and accuracy, enabling our system to respond swiftly to on-target pay-load object drops with minimal latency and high frame rate.

3.2.5 Integration

The Object Detection Module seamlessly integrates with other components of the system, such as the Automatic Alert Generator, to provide a comprehensive solution for detecting and responding to on-target payload object drops in real-time. Through efficient communication and data sharing, the module enhances the overall effective-ness of the system in safeguarding cargo and ensuring operational safety.

By leveraging the advanced capabilities of the YOLO-NAS model, our system empowers stakeholders in the logistics, manufacturing, and security sectors with a robust solution for addressing the challenges posed by payload object drops.

3.3 Alert Generation Module

3.3.1 Introduction to Gmail API

In the context of the proposed system, efficient and timely alert generation is imperative for ensuring swift responses to detected incidents. To streamline this process, the system leverages the Gmail API, a powerful tool provided by Google that enables seamless integration of email functionality into applications and services.

3.3.2 Overview

The Gmail API offers a comprehensive set of features for programmatically accessing and managing Gmail accounts, including the ability to send, receive, and manage emails. It provides developers with a standardized interface to interact with Gmail accounts, eliminating the need for manual email management and enabling automated communication within applications.

3.3.3 Integration with Object Detection Module

Within the context of the system, the Gmail API is used in the Alert Generation Mod-ule, where it plays a pivotal role in notifying relevant stakeholders about detected incidents, such as the off-target payload object drops. Upon detection of an incident by the Object Detection Module, the alert flag in the Alert Generation Module toggles and initiates the alerting process by invoking the Gmail API to send notification emails to predefined recipients, including designated personnel, administrators, or relevant authorities.

3.3.4 Key Features and Functionality

- Email Composition: The Gmail API allows for the programmatically composition of email messages, enabling the Alert Generation Module to craft customized alert notifications containing pertinent information about the detected incident, including timestamp, location, and severity.

- Recipient Management: Through the Gmail API, the Alert Generation Module can dynamically manage recipient lists, ensuring that notifications are sent to the appropriate individuals or groups based on predefined criteria or escalation paths.

- Real-time Delivery: Leveraging the real-time capabilities of the Gmail API, alerts are delivered promptly to recipients, facilitating swift responses to detected incidents and minimizing response times.

3.3.5 *Security Considerations*

The Gmail API incorporates robust security measures to safeguard sensitive information and ensure secure communication. OAuth 2.0 authentication is employed to authenticate and authorize access to Gmail accounts, mitigating the risk of unauthorized access or misuse of email functionality.

4. DETECTION MODEL RESULTS

The results obtained after automated and manual testing and running the inference on sample images, and videos showed great accuracy and high-speed inference. Training as well as test results are shown below:

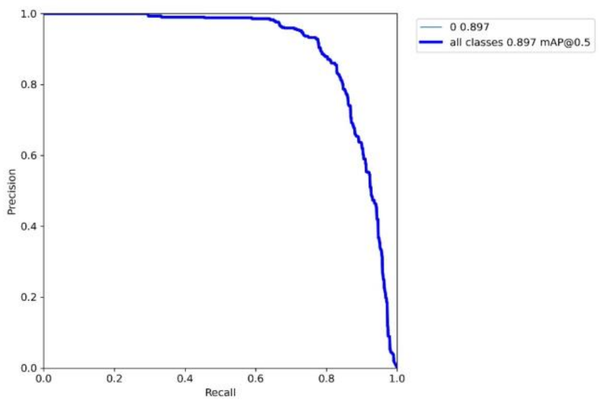


Figure 8. Precision-Recall Curve

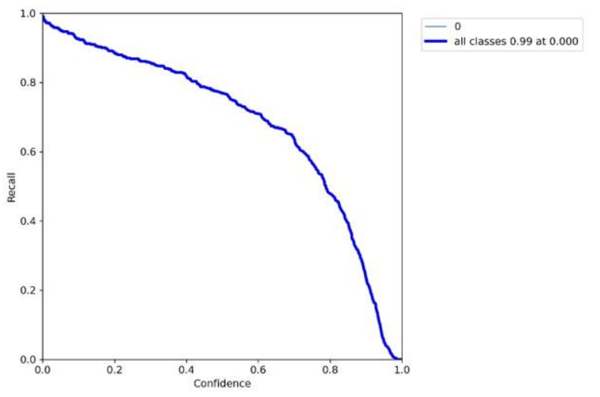


Figure 9. Recall Curve

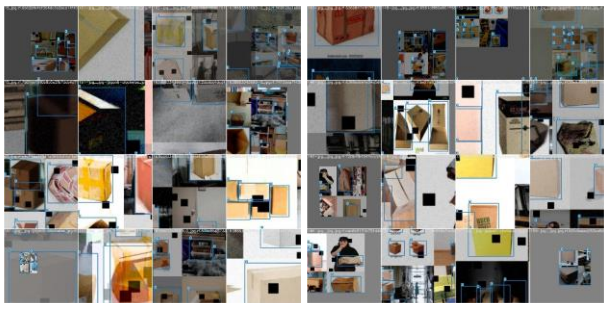


Figure 10. Batched Test Images

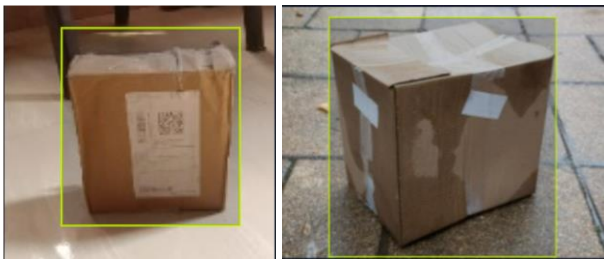


Figure 11. Live Test Images

5. SUMMARY AND FUTURE SCOPE

The "Detection of On-Target Payload Object Drop and Automatic Alert Generator" system represents a significant advancement in the field of cargo security and operational efficiency within the logistics, manufacturing, and security sectors. Through the integration of cutting-edge technologies such as the YOLO-NAS deep learning model, the NVidia DGX A100 system, and the Gmail API, the system addresses critical challenges associated with the timely detection and response to on-target payload object drops.

In summary, this research paper has presented a comprehensive overview of the system architecture, implementation details, and performance evaluation of key modules including the Object Detection Module, Target Provisioning System, and Alert Generation Module. By leveraging state-of-the-art techniques in object detection, image processing, and communication, the system demonstrates remarkable capabilities in enhancing cargo security, minimizing operational disruptions, and ensuring worker safety.

Looking ahead, the prospects of this research project are multifaceted and promising. Firstly, ongoing advancements in deep learning models and high-performance computing hardware are expected to further enhance the accuracy and efficiency of the object detection module, enabling the system to handle complex scenarios with even greater precision. The integration of additional sensors and data sources, such as RFID tags, GPS trackers, and environmental sensors, holds immense potential for enriching the contextual information available to the system. This expanded data landscape can enable more nuanced decision-making processes and enhance the system's ability to adapt to dynamic operational environments. Furthermore, the utilization of advanced communication protocols and platforms beyond email, such as real-time messaging systems or integration with centralized incident management platforms, could further streamline the alert generation and response process, facilitating faster and more coordinated responses to detected incidents.

The scalability and adaptability of the system make it well-suited for deployment in diverse industries beyond logistics and manufacturing, including critical infrastructure protection, public safety, and emergency response.

In conclusion, the "Detection of On-Target Payload Object Drop and Automatic Alert Generator" system represents a significant step forward in enhancing cargo security and operational efficiency. By embracing emerging technologies and exploring novel integration possibilities, the system is poised to continue evolving, offering tangible benefits to a wide range of industries and stakeholders in the foreseeable future.

6. REFERENCES

[1] Slyusar, Vadym, Mykhailo Protsenko, Anton Chernukha, Vasyl Melkin, Oleh Biloborodov, Mykola Samoilenko, Olena Kravchenko, Halyna Kalynychenko, Anton Rohovyi, and Mykhaylo Soloshchuk. "Improving The Model Of Object Detection On Aerial Photographs And Video In Unmanned Aerial Systems." Eastern-European Journal of Enterprise Technologies 1, no. 9 (2022): 115.

[2] Kalli, SivaNagiReddy, T. Suresh, Aruchamy Prasanth, T. Muthumanickam, and K. Mohanram. "An effective motion object detection using adaptive background modeling mechanism in video surveillance system." Journal of Intelligent & Fuzzy Systems 41, no. 1 (2021): 1777-1789.

[3] Hakim, Muhammad Luthfi, Sigit Yatmono, Ariadie Chandra Nugraha, and Moh Khairudin. "AUTONOMOUS QUADCOPTER WITH IMAGE OBJECT DETECTION METHOD AS A SENDER OF ASSISTANCE FOR COVID-19 PATIENTS." International Journal of Mechatronics and Applied Mechanics 10 (2021): 18-23.

[4] Sewell, James, Theo van Niekerk, Russell Phillips, Paul Mooney, and Riaan Stopforth. "Vision-Based Object Detection and Localization for Autonomous Airborne Payload Delivery." In CONTROLO 2020: Proceedings of the 14th APCA International Conference on Automatic Control and Soft Computing, July 1-3, 2020, Bragança, Portugal, pp. 602-615. Springer International Publishing, 2021.

[5] Sahin, Oyku, and Sedat Ozer. "Yolodrone: Improved yolo architecture for object detection in drone images." In 2021 44th International Conference on Telecommunications and Signal Processing (TSP), pp. 361-365. IEEE, 2021.

[6] Jha, Sudan, Changho Seo, Eunmok Yang, and Gyanendra Prasad Joshi. "Real time object detection and trackingsystem for video surveillance system." Multimedia Tools and Applications 80 (2021): 3981-3996.

[7] Krišto, Mate, Marina Ivasic-Kos, and Miran Pobar. "Thermal object detection in difficult weather conditions using YOLO." IEEE access 8 (2020): 125459-125476.

[8] Yang, Bo, Xuelin Cao, Chau Yuen, and Lijun Qian. "Offloading optimization in edge computing for deep-learning-enabled target tracking by internet of UAVs." IEEE Internet of Things Journal 8, no. 12 (2020): 9878-9893.

[9] Zhang, Zhong, Eric Becker, Roman Arora, and Vassilis Athitsos. "Experiments with computer vision methods for fall detection." In Proceedings of the 3rd international conference on pervasive technologies related to assistive environments, pp. 1-4. 2010.

[10] Wu, Shicheng, Rui Li, Yingjing Shi, and Qisheng Liu. "Vision-based target detection and tracking system for a quadcopter." IEEE Access 9 (2021): 62043-62054.

[11] Akbari, Younes, Noor Almaadeed, Somaya Al-Maadeed, and Omar Elharrouss. "Applications, databases and open computer vision research from drone videos and images: a survey." Artificial Intelligence Review 54 (2021): 3887-3938.

[12] Rabah, Mohammed, Ali Rohan, Muhammad Talha, Kang-Hyun Nam, and Sung Ho Kim. "Autonomous vision-based target detection and safe landing for UAV." International Journal of Control, Automation and Systems 16 (2018): 3013-3025.

[13] Boudjit, K., and Chevif Larbes. "Detection and implementation autonomous target tracking with a Quadrotor AR. Drone." In 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 2, pp. 223-230. IEEE, 2015.

[14] Vadduri, Aditya, Anagh Benjwal, Abhishek Pai, Elkan Quadros, Aniruddh Kammar, and Prajwal Uday. "Precise Payload Delivery via Unmanned Aerial Vehicles: An Approach Using Object Detection Algorithms." arXiv preprint arXiv:2310.06329 (2023).

[15] Yao, Chun-Han, Chen Fang, Xiaohui Shen, Yangyue Wan, and Ming-Hsuan Yang. "Video object detection via object-level temporal aggregation." In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, pp. 160-177. Springer International Publishing, 2020.

[16] Kumar, S., Singh, A. K., Bhushan, S., & Vashishtha, A. (2022, September). Polarities Inconsistency of MOOC Courses Reviews Based on Users and Sentiment Analysis Methods. In Proceedings of 3rd International Conference on Machine Learning, Advances in Computing, Renewable Energy and Communication: MARC 2021 (Vol. 915, p. 361). Springer Nature.

[17] Kumar, S., Singh, A.K., Bhushan, S., Vashishtha, A. (2022). Polarities Inconsistency of MOOC Courses Reviews Based on Users and Sentiment Analysis Methods. In: Tomar, A., Malik, H., Kumar, P., Iqbal, A. (eds) Proceedings of 3rd International Conference on Machine Learning, Advances in Computing, Renewable Energy and Communication. Lecture Notes in Electrical Engineering, vol 915. Springer, Singapore. https://doi.org/10.1007/978-981-19-2828-4_34

[18] Kumar, S., Singh, A.K., Bhushan, S., Kumar, P., Vashishtha, A. (2022). A Comparative Study on Sentiment Analysis of Uber and Ola Customer Reviews Based on Machine Learning Approaches. In: Pundir, A.K.S., Yadav, N., Sharma, H., Das, S. (eds) Recent Trends in Communication and Intelligent Systems. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-19-1324-2_7

International Journal of Computer Applications Technology and Research
Volume 13–Issue 04, 07 - 17, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1304.1002

Malware Attacks Classification Model

Ogwuche I.T.
Joseph Sarwuan Uni
Makurdi Benue State

Dr. Gbaden T.
Joseph Sarwuan Uni
Makurdi Benue State

Dr Ogala E
Joseph Sarwuan Uni
Makurdi Benue State

Yugh M.S
Joseph Sarwuan Uni.
Makurdi Benue State

Ekoja P
Ben State Poly
Benue State Nigeria

Abstract: This study developed a model to classify attacks in a digital economy system using Random Forest and support vector machine. The Rational Unified process research methodology was used to develop the model. It was implemented using the spyder notebook development environment and using Python programming language version 3.7. In this research, experiments were conducted to check the performance of the models based on the accuracy, precision, recall rate, and F1 – Score. From the results achieved, the classification metrics shows that the Random **Forest** Classifier scored 98.9% in accuracy, precision, recall rate, and F1 – Score. The classification metrics show that the Support Vector Machine scored 98.9% in accuracy, precision, recall rate, and F1 – Score. The experimental result implies that Random Forest Classifier and Support Vector Machine Classifier scored the same in performance when compared. This research contributed to the enhancement of threat classification and made a proper decision(s) as to the rate of occurrence of specific types of threats using Random Forest and Support Vector Machine.

Keywords: Threats, attacks, Digital Economy, Model

1. INTRODUCTION

The rapid spread of the use of technology aided by the internet has brought about so many transformations to our economic and social lives. This transformation is now termed as digital economy. Digital Economy allows and enhances trade of goods and services to be executed via electronic commerce on the Internet (Bukht and Heeks, 2017). Nigeria's cashless policies have accelerated internet use, enabling a digital economy. However, the digital economy faces security threats like smart threats, ransomware, and malware attacks, with malware being the most prevalent threat (Erdal and Milad, 2019).

Supervised machine learning is self-learning that depends upon labels from the data. A supervised learning algorithm identifies abnormal indications of deep threats concerning labeled outcomes as opposed to unsupervised learning (Hamad *et al.*, 2019).

Cyber security involves policies, processes, technologies, and techniques to protect computing resources, networks, software programs, and data from attacks. Tools like firewalls, antivirus software, intrusion detection systems, and IPS prevent attacks. However, the increasing number of internet-connected systems increases the risk of attacks. As attackers become more sophisticated, they develop zero-day exploits and malware that evade security measures

These vulnerabilities if not addressed will cause huge financial losses in the digital economy, therefore, against this backdrop, the research has developed a model that classifies attacks arising from vulnerabilities in systems and trains the model to detect these attacks.

This research classifies the topologies of attacks that threaten the security of digital services and recommends solutions for the protection of the digital economy and its services. The research used a secondary dataset from an online data source to train the model

2. LITERATURE REVIEW

Hansen, (2016) implemented a novel Monte Carlo tree search algorithm to forecast the presence of active malware in the training dataset. To forecast the new form of malware as the test data, a new search model is created in this research and applied to the active malware attacks. Using the search technique, it is difficult to forecast the high number of active malware.

Aghaeikheirabady,(2014) implemented different malware attack detection models on the training dataset using a comparison of the information in the user space memory data structures to expedite information extraction and ensure accuracy. In this method, malware artifacts related to registry modifications as well as calls to library files and operating system routines using descriptions of memory structures are recovered. Following an evaluation of the retrieved features, samples are categorized using the chosen attributes. The best outcomes show a 98% detection rate and a 16% false positive rate, demonstrating the efficiency of the suggested behavior extraction method. In this work, the data imbalance problem affects the true positive rate and error rate.

Selvi, (2019) offers a machine learning method that uses Random Forest to identify algorithmically produced domains only based on the lexical properties of the domain names. They particularly recommend combining other statistics gleaned from the domain name with masked N-grams. Additionally, they offer a dataset created for experimentation that contains domain names that were randomly and artificially produced from various malware families. They additionally group these families based on the domain creation algorithm used. As a result, masked N-grams offer detection accuracy that is comparable to that of other methods now in use while also offering substantially better performance. This approach can eventually be expanded to big, imbalanced datasets.

Takase *et al*.(2019) proposed employing values taken out of the CPU as part of a malware detection procedure. By utilizing processor information, they offload the malware detection technique to hardware while reducing the demand on the hardware's resources. Virtual machine QEMU was used to create a prototype of the method described. A demonstration on how the suggested technique can distinguish between malicious and good applications using processor information and can also identify malware variants that belong to the same family was done. However, evaluation results show that it is possible Trace Data could be incorrectly categorised based on the combination of training programs.

Ijaz*et al*. (2019) used dynamic malware analysis and different feature combinations. The various combinations are produced via APIs, Summary Data, DLLs, and RegKey Changed. Dynamic malware analysis is done with the help of the adaptable and accurate Cuckoo Sandbox. Using PEFILE, more than 2300 features are statically collected from binary malware and 92 features are dynamically extracted from malware analysis. From 10000 benign files and 39,000 dangerous binaries, static features are retrieved. In the Cuckoo Sandbox, 800 benign files and 2200 malware files are dynamically examined, and 2300 characteristics are extracted. Static analysis is 99.36% accurate, compared to 94.64% for dynamic malware analysis. Analysis of dynamic malware is ineffective due to malware's cunning and intelligent behavior. Due to regulated network behavior and limited network access, dynamic analysis has some restrictions and cannot be fully examined.

Foley *et al.* (2019) identified coupled attacks against two well-known objective function (OF). OF is one of the important features of routing protocol for low-power and lossy networks (RPL). They investigate their vulnerability assessments utilizing various simulated situations and machine learning algorithms. To do this, they developed a unique IoT dataset based on network and power parameters, which is implemented as a component of an RPL IDS/IPS system to improve information security. Regarding the results that were recorded, their machine learning approach is effective at spotting combination attacks against two well-known RPL OFs based on the power and network metrics, where MLP and RF algorithms are the most successful classifier deployment for both single and ensemble models.

Ieracitano *et al.* (2020), presented a novel statistical analysis-driven intelligent intrusion detection system (IDS). To extract more optimal, strongly correlated characteristics, the proposed IDS specifically blend data analytics and statistical methods with current developments in machine learning theory. By comparing it to the benchmark National Security Database (NSL-KDD), the suggested IDS are evaluated. Comparative experimental results demonstrate that, when compared to conventional deep and shallow machine learning and other recently proposed state-of-the-art methodologies, the planned statistical analysis and stacked Autoencoder (AE) based IDS obtain greater classification performance.

He and Kim,(2019) researched how well Convolutional Neural Networks (CNN) defends against the injection of redundant API. By converting malware files into picture representations and classifying the image representation with CNN, they created a malware detection system SPP or spatial pyramid pooling layers used to create CNN to handle input of different sizes. By assessing the performance of this system on both unaltered data and hostile data with redundant API injection, they assess the usefulness of SPP and picture color space (grayscale/RGB). Results indicate that greyscale imaging is effective against redundant API injection whereas naïve SPP implementation is problematic owing to memory limitations.

Ren*et al*.(2020) presented hierarchical clustering with a decision tree model, a hybrid malware detection model on a limited malware dataset. It has difficulty in identifying multi-class attacks on real-time malware dataset

Alhanahnah*et al*.(2019) transformed the program OpCodes into a vector space and applied fuzzy and fast fuzzy pattern tree algorithms for malware identification and categorization. Particularly for the fast fuzzy pattern tree, a high level of accuracy was achieved with a manageable run-time. Edge computing malware detection and categorization methods become more effective as a result of the robust use of both feature extraction and fuzzy classification. However, it applies to samples built on ARM and there are not enough samples in the dataset.

Given the various literature reviewed, the authors' works addressed the following issues:

i. Investigations on integrated risk and damage assessment in the digital economy using fuzzy approaches.
ii. Anomaly Detection on IOT system based on Random Forest.
iii. A hybrid malware detection model using hierarchical clustering with a decision tree to identify multi-class attacks on real-time malware datasets.
iv. Creating strategies for distinguishing between signatures for IOT malware.

Due to the unbalanced nature of datasets used in the reviewed works, problems of true positive rates and error rates were encountered. Also, the authors did not take into account the different categories of digital economy threats. It is as a result of the aforementioned drawbacks that this research adopts Random Forest and Support Vector Machine algorithms to classify digital economy threats into eight categories using a balanced dataset to compensate for the large true positive recorded in the reviews above

3. METHODOLOGY

The methodology adopted in this research work is Rational Unified Process (RUP). RUP is an iterative software development process framework that makes heavy use of Unified Modeling Language (UML) in its phases.

Support Vector Machine and Random Forest algorithms are used to develop a model for classifying malware attacks in the activities of the digital economy in Nigeria. A data set in the form of .csv file was used to train and test the model. The results helped evaluate how malware attacks can be prevented.

3.1 Analysis of Existing System

Malware classification and detection is an emerging prospect that seeks to resolve cybersecucurity concerns arising from the activities of fraudulent persons. Some research works such as Ren *et al (2020)* have been carried out and a hybrid malware detection model was built using hierarchical clustering with a decision tree model however; it had difficulty in identifying multi-class attacks on real-time malware dataset. In Nigeria, malware incidents are not detected and recorded real-time thereby creating room for malware incidences to go undetected and unabated. The common practice is the use of antivirus software which is inadequate against the evolving nature of malware because malware takes advantage of security vulnerabilities in hardware and network traffic.

3.2 Analysis of The Proposed System

Random Forest and Support Vector Machine was the machine learning algorithms used in the training and testing of our model

Random Forest (RF): is an algorithm for ensemble machine learning. Several decision trees (DT) are combined into a forest to operate this algorithm. DT is a frequently used data mining ML method that is able to solve classification and regression problems, it is highly suited for resolving classification tasks. This algorithm categorizes a population into branch-like segments that construct a reversed tree with a root node, internal nodes, and leaf nodes. The ML method is non parametric and can competently deal with large, complex, and complicated data without imposing a complex parametric structure. DT learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a DT. Learned trees can also be shown as sets of if-then rules to improve human legibility. The DT model makes analysis based on three basic nodes, namely:
• Root node: principal node based on this node all-other nodes functions.
• Interior node: handles various features or attributes of the dataset.
• Leaf node: represent the outcome of each test, i.e., the class of the dependent variable in classification problems.
The DT algorithm divides the data into two or more analogous sets based on the most significant indicators. The entropy of each feature is computed, and then the dataset is divided, with predictors having the minimum entropy. The formula for calculating the entropy of an attribute is shown in Eq. (1).
Entropy s ( ) c ∑ = − i = 1 p log 2 p i i (1)
Where in Eq. (1), c is the total number of classes and the probability of samples belonging to a class at a given node can be denoted as p i
The class with the highest votes becomes the model's forecast in the RF algorithm, which generates a class expectation from each distinct tree. The likelihood of improved accuracy increases with the number of trees in an RF classifier. While it is more effective at classification tasks and can overcome the drawbacks of DT and other ML approaches, like overfitting and missing values, it can be utilized for both regression and classification tasks (Ekle *et al,* 2023)
A supervised machine learning approach that may be applied to both regression and classification problems is the support vector machine (SVM). With support vector machines, data points are plotted as points in an n-dimensional space, where n is the number of features. The value of each feature is represented by a specific coordinate. By identifying the ideal hyperplane that best distinguishes the two classes, the classes are classified into their appropriate categories.

The SVM Classifier:

The hypothesis function is defined as:

$$h\,(x_i) = \begin{cases} +1 & if\ w.x + b \geq 0 \\ -1 & if\ w.x + b < 0 \end{cases}$$

Class +1 will be assigned to the point that is above or on the hyperplane, while class -1 will be assigned to the point that is below the hyperplane.
To compute the (soft-margin) SVM classifier, one must minimize the following expression:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(w \cdot x_i - b)\right)\right] + \lambda\|$$

Since selecting a small enough value for lambda produces the hard-margin classifier for linearly classifiable input data, we concentrate on the soft-margin classifier.(Kecman, Vojislav, 2005)

3.3. System Design

The proposed system uses the support vector machine algorithm and random forest algorithm to classify and detect threats from nine families of attacks identified from security vulnerability in system and network traffic.

3.3.1 Architectural Model of the System

The architecture of the model consists of: –

i. data collection-data is from an online data source https://www.kaggle.com in the form of .csv file.
ii. Data Extraction- The dataset is UNSW-NB15 (computer network security dataset released in 2015).Nine families of attack namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Reconnaissance, Shellcode, and Worms are extracted. The total number of records is two hundred and fifty seven thousand, six hundred and seventy three (257,673) The number of records in the training set is 175,341 records and the testing set is 82,332 records from different types of attack and normal.
iii. The data is normalized and then fed into the Random Forest and Support vector classifier to train and test the model. The results of the model will be displayed and evaluated based on the following parameters: F1, recall, accuracy, and precision. The architectural model is given in Figure 1. The architectural model is broken down into five phases as seen in figure 1.

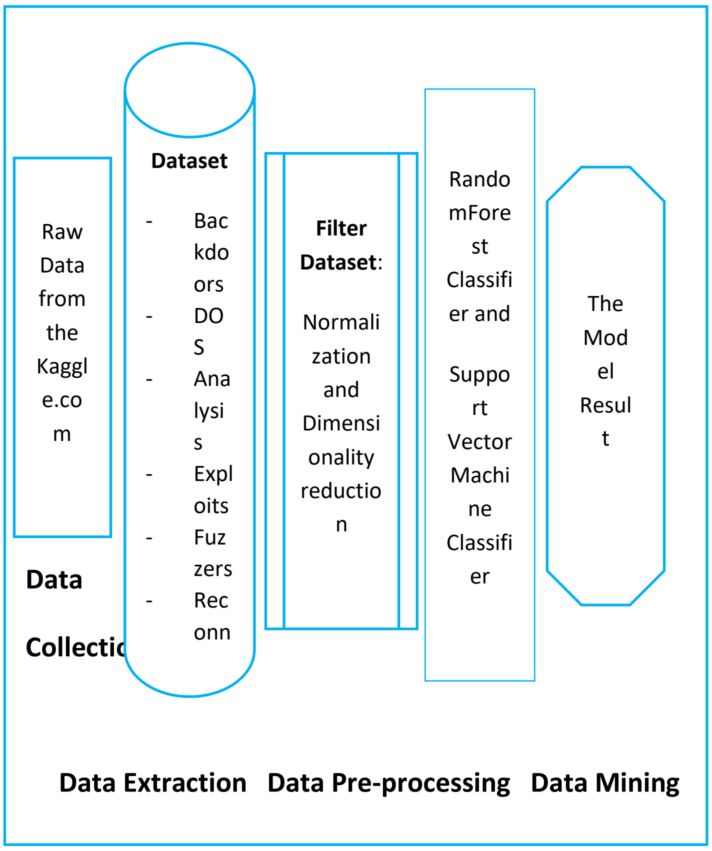


Figure 1: Architectural Model

3.3.2 Activity Diagram

is a system design tool that describes all the activities of a system from start to finish. The small black circle depicts the beginning of the activities of the system as seen in figure 2; the arrow is a pointer to the next activity. The machine learning algorithms in this study are random forest and support vector machine. There are two algorithms in this activity, so the arrow has to be split in two different activities support vector machine and random forest, so it requires an object called the fork node. The fork node in an activity diagram is used specify where a single activity is divided into two or more activity. The small, dark inner shaded circle with an outer unshaded circle marks the end of all activity. The activity diagram is shown in figure 2:

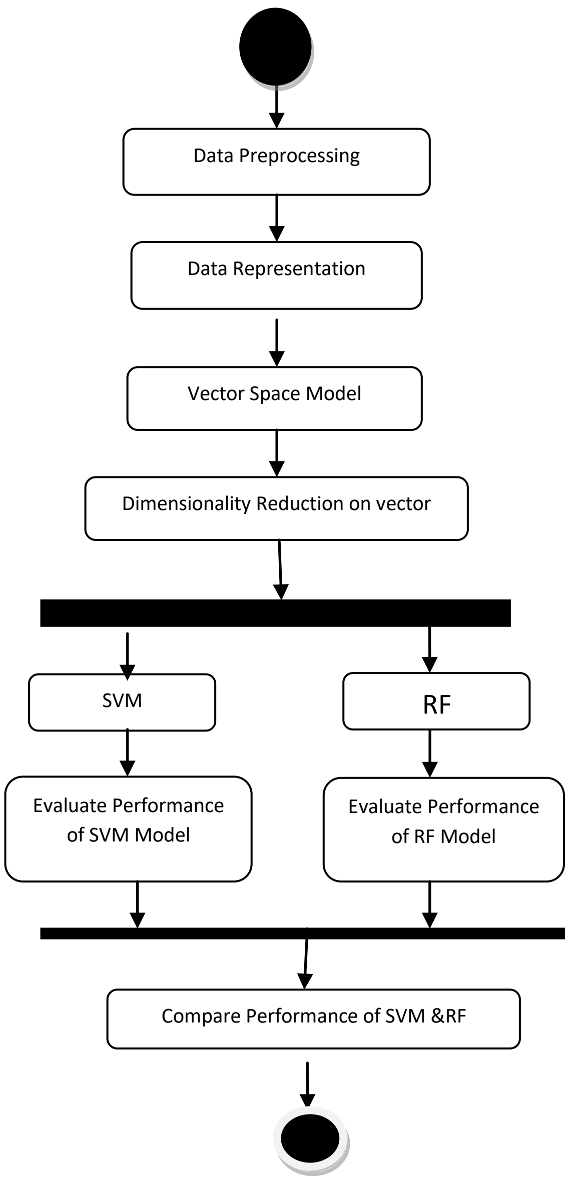

Data Preprocessing
Data Representation
Vector Space Model
Dimensionality Reduction on vector
SVM
RF
Evaluate Performance of SVM Model
Evaluate Performance of RF Model
Compare Performance of SVM &RF

Figure 2: Activity Diagram

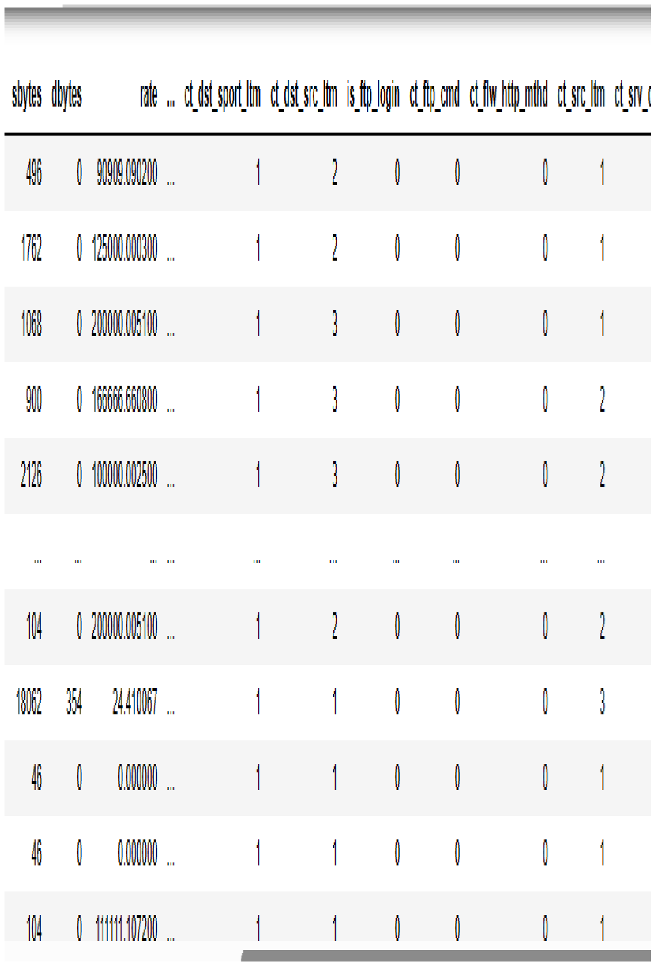
4. IMPLEMENTATION AND RESULTS

The threat classification technique is implemented using Python programming environment version 3.7. This chapter describes the nature of the datasets, data preprocessing, experimental set-up, experiment evaluation, and finally experimental results.

4.1 Description of Datasets

The experiment in this research was conducted on raw network packets of the data set created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviours. This data set has nine families of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The data set utilised 49 features with the class label. A partition from this data set is configured as a training set and testing set respectively. The number of records in the training set is 175,341 records and the testing set is 82,332 records. Table 1 represent an extract of the data set of digital economy threats used for the classification problem.

Table 1: An Extract of the Data set



sbytes	dbytes	rate	...	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd	ct_flw_http_mthd	ct_src_ltm	ct_srv_
496	0	90909.090200	...	1	2	0	0	0	1	
1762	0	125000.000300	...	1	2	0	0	0	1	
1068	0	200000.005100	...	1	3	0	0	0	1	
900	0	166666.660800	...	1	3	0	0	0	2	
2126	0	100000.002500	...	1	3	0	0	0	2	
...	...	...	...	...	...	...	...	...	...	
104	0	200000.005100	...	1	2	0	0	0	2	
18062	354	24.410067	...	1	1	0	0	0	3	
46	0	0.000000	...	1	1	0	0	0	1	
46	0	0.000000	...	1	1	0	0	0	1	
104	0	111111.107200	...	1	1	0	0	0	1	

Table 2: Transposed Data set

Index	Rate	Attack_ Cat	Label
2709	16.4677	7	1
2710	29.8079	3	1
2711	125000	3	1
2712	58.466	2	1
2713	12.5000	4	1
2714	41.887	4	1
2715	3333333	7	1
2716	111111	4	1
2717	111111	7	1
2718	125000	7	1
2719	125000	2	1
2720	111111	3	1
2721	105.548	2	1
2722	333333	2	1

4.2 Data Preprocessing

The preprocessing on data set is done by removing the unwanted columns that are present in the data set. The columns named ‘rate’, ‘attack_cat’ and ‘label’are the features of interest; every other aredropped as it’s not needed. The ‘attack_cat’ column contains nominal values that cannot be used to train the models of interest, so we have converted the nominal values in the ‘attack_cat’ column into numeric values using the “One Hot Encoder” algorithm. Now, the data is ready to work with as seen in Table 2. Figure 6 indicates heat map for visualizing missing values in the dataset. Figure 7 indicates the dataset's correlation matrix. This matrix clarifies that the ‘label’ attribute is dependent on the ‘attack_cat’and‘rate’ attributes

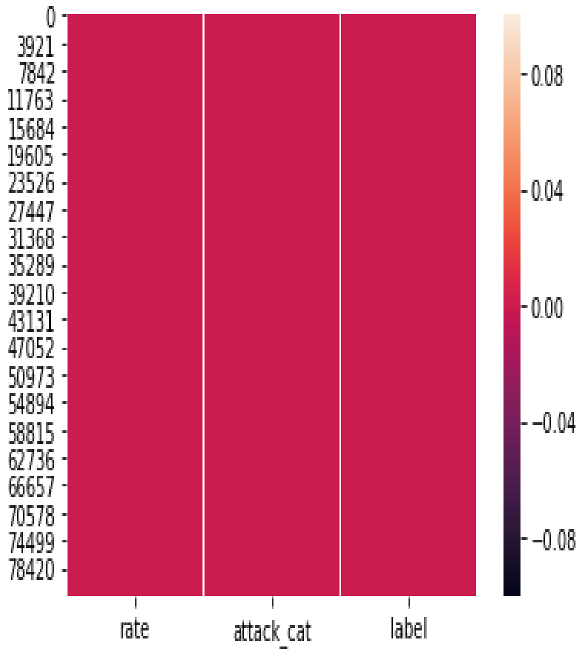


Figure 6: Heat map of the dataset

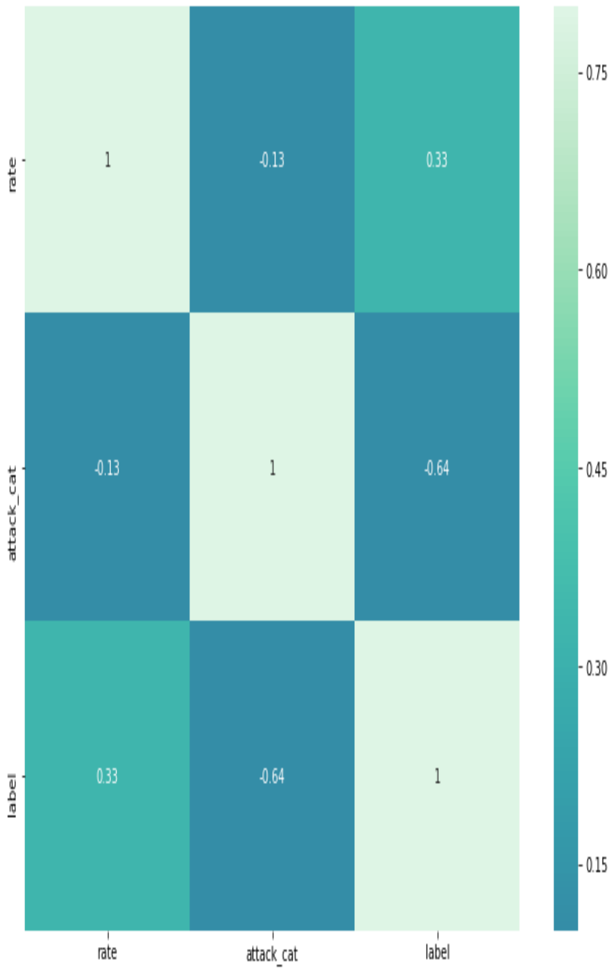


Figure 7: Dataset Correlation Matrix

4.2 Evaluation of Experiments

In this research, experiments were conducted to check the performance of the models based on the accuracy, precision, recall rate and F1 - Score of the models. The results of the experiments are recorded in Table 3 and Table 4 below. In Table 3, the classification metrics shows that the Random Forest Classifier scored 98.9% in accuracy, precision, recall rate, and F1–score. The classification metrics in Table 4 shows that Support Vector Machine scored 98.9% in accuracy, precision, recall rate, and F1 – Score. The experimental result implies that Random Forest Classifier and Support Vector Machine Classifier scored the same in performance when compared.

Table 3: Random Forest Classification Reports

Random Forest Errors:	0			
The Matthews correlation coefficient is:	1.0			
	Precision	Recall	f1-score	support
0	0.98	0.98	0.98	11147
1	0.98	0.98	0.98	13553
Accuracy			0.98	24700
macro avg	0.98	0.98	0.98	24700
weighted avg	0.98	0.98	0.98	24700

Table 4: Support Vector Machine Classification Reports

Support Vector Machine Errors:	0			
The Matthews correlation coefficient is:	1.0			
	Precision	Recall	f1-score	support
1	0.98	0.98	0.98	1903
Accuracy			0.98	1903
macro avg	0.98	0.98		
weighted avg	0.98	0.98		

5. DISCUSSIONS

5.1 Experimental Results and Discussions

The results of the experiment conducted in table 2 are shown and discussed in figures 8,9,10,11 and 12 respectively

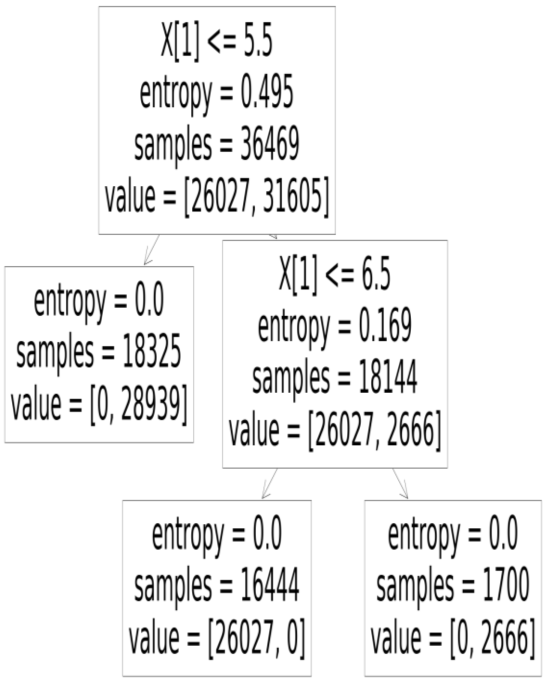


Figure 8: First Decision Tree

Figure 8 shows a decision tree with root node that consists of X [1] <= 5.5, entropy = 0.495, samples = 36469, and value = [26027, 31605]. Since the entropy value (0.495) is relatively high indicating uncertainty in our dataset, the root node is splitted into left (if X[1] <= 5.5 is True) and right (if X[1] <= 5.5 is False) nodes. The left intermediate node is now a leaf node (entropy = 0.0, samples = 18325 and value = [0, 28939]) and cannot be splitted further since it has zero entropy value which indicates certainty in our dataset. The right intermediate node (X[1] <= 6.5, entropy = 0.169, samples = 18144 and value = [26027, 2666]) will be splitted further into left (if X[1] <= 6.5 is True) and right (if X[1] <= 6.5 is False) nodessince it has entropy value (0.169) which indicates uncertainty in our dataset. The left (entropy = 0.0, samples = 16444 and value = [26027, 0]) and right (entropy = 0.0, samples = 1708 and value = [0, 2666]) nodes are now leaf nodes and cannot be splitted further since it has zero entropy values which indicates certainty in our dataset. Since the entropy values of all the leaf nodes are zero, it implies that information is 100% present and the path of the deepest leaf node rightmost is the best decision path since the node has the least number of samples (1700).

=

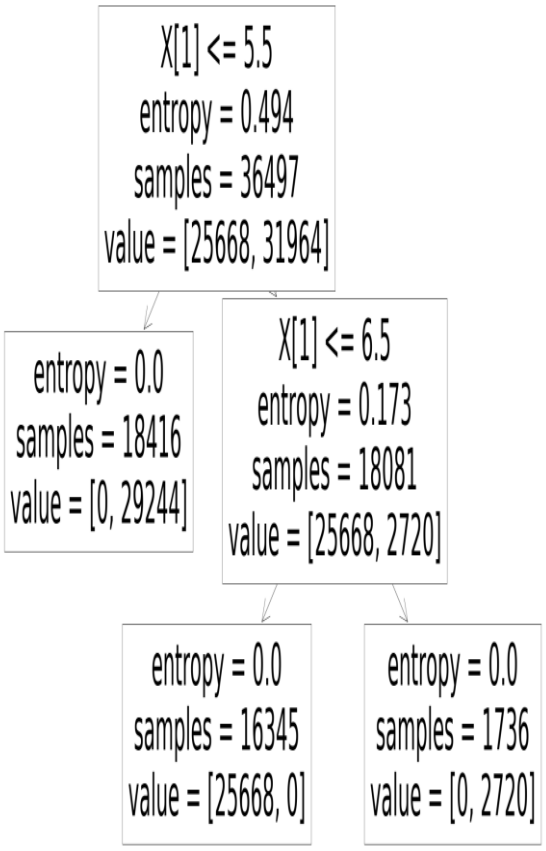


Figure 9: Second Decision Tree

Figure 9 shows a decision tree with root node which consists of X[1] <= 5.5, entropy = 0.494, samples = 36497 and value = [25668, 31964]. Since the entropy value (0.494) is relatively high indicating uncertainty in our dataset, the root node is splitted into left (if X[1] <= 5.5 is True) and right (if X[1] <= 5.5 is False) nodes. The left intermediate node is now a leaf node (entropy = 0.0, samples = 18416 and value = [0, 29244]) and cannot be splitted further since it has zero entropy value which indicates certainty in our dataset. The right intermediate node (X[1] <= 6.5, entropy = 0.173, samples = 18081 and value = [25668, 2720]) will be splitted further into left (if X[1] <= 6.5 is True) and right (if X[1] <= 6.5 is False) nodes since it has entropy value (0.173) which indicates uncertainty in our dataset. The left (entropy = 0.0, samples = 16345 and value = [25668, 0]) and right (entropy = 0.0, samples = 1736 and value = [0, 2720]) nodes are now leaf nodes and cannot be splitted further since it has zero entropy values which indicates certainty in our dataset. Since the entropy values of all the leaf nodes is zero, it implies that information is 100% present and the path of the deepest leaf node rightmost is the best decision path since the node has the least number of samples (1736).

Figure 10: Third Decision Tree

Figure 10 shows a decision tree with a root node that consists of X[1]<= 5.5, entropy = 0.494, samples = 36378, and value = [25707, 31925]. Since the entropy value (0.494) is relatively high indicating uncertainty in our dataset, the root node is splitted into left (if X[1] <= 5.5 is True) and right (if X[1] <= 5.5 is False) nodes. The left intermediate node is now a leaf node (entropy = 0.0, samples = 18462, and value = [0, 29204]) and cannot be splitted further since it has zero entropy value which indicates certainty in our dataset.The right intermediate node (X[1] <= 6.5, entropy = 0.173, samples = 17916 and value = [25707, 2721]) wassplitted further into left (if X[1] <= 6.5 is True) and right (if X[1] <= 6.5 is False) nodessince it has entropy value (0.173) which indicates uncertainty in our dataset. The left (entropy = 0.0, samples = 16208 and value = [25707, 0]) and right (entropy = 0.0, samples = 1708 and value = [0, 2721]) nodes are now leaf nodes and cannot be splitted further since it has zero entropy values which indicates certainty in our dataset. Since the entropy values of all the leaf nodes are zero, it implies that information is 100% present and the path of the deepest leaf node rightmost is the best decision path since the node has the least number of samples (1708).

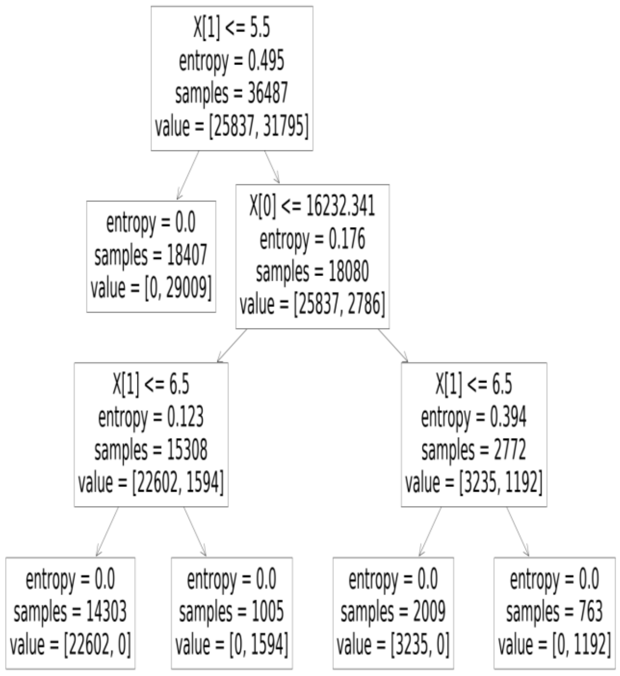


Figure 11: Fourth Decision Tree

Figure 11 shows a decision tree with root node that consists of X[1] <= 5.5, entropy = 0.495, samples = 36487, and value = [25837, 31795]. Since the entropy value (0.495) is relatively high indicating uncertainty in our dataset, the root node is splitted into left (if X[1] <= 5.5 is True) and right (if X[1] <= 5.5 is False) nodes. The left intermediate node is now a leaf node (entropy = 0.0, samples = 18407 and value = [0, 29009]) and cannot be splitted further since it has zero entropy value which indicates certainty in our dataset. The right intermediate node (X[0] <= 16232.341, entropy = 0.176, samples = 18080 and value = [25837, 2786]) wassplitted further into the left (if X[1] <= 6.5 is True) and right (if X[1] <= 6.5 is False) nodessince it has entropy value (0.176) which indicates uncertainty in our dataset. Repeat the split process until all nodes become leaf nodes with zero entropy values indicating that information is 100% present and the path of the deepest leaf node rightmost is the best decision path since the node has the least number of samples (763).

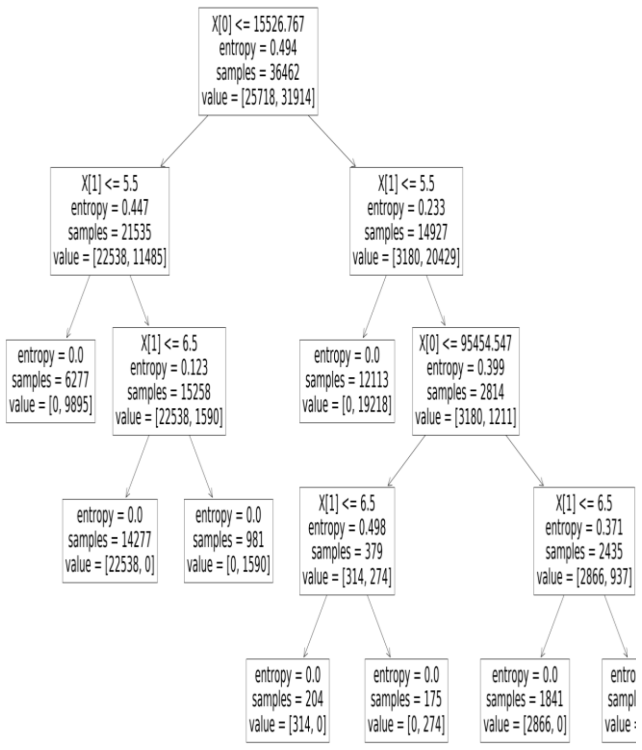


Figure 12: Fifth Decision Tree

Figure 12 shows a decision tree with a root node that consists of X[0] <= 15526.767, entropy = 0.494, samples = 36462, and value = [25718, 31914]. Since the entropy value (0.494) is relatively high indicating uncertainty in our dataset, the root node is splitted into left (if X[1] <= 5.5 is True) and right (if X[1] <= 5.5 is False) nodes. Repeat the split process until all nodes become leaf nodes with zero entropy values indicating that information is 100% present and the path of the deepest leaf node with samples = 175 is the best decision path since the node has the least number of samples.

6. CONCLUSION AND RECOMMENDATION(S)

6.1 Conclusion

This research motivation is to enhance threat classification in the digital economy and make a proper decision (s) as to the rate of occurrence of specific types of threats using Random Forest and Support Vector Machine. Numerous experiments were conducted on digital threat datasets using Random Forest and Support Vector Machine algorithms with different decision trees showing the best decision path. The experiment was conducted in Python version 3.7 programming environment. The experiment indicated 98.9% performance of the proposed classification techniques on the training dataset. Based on the analysis conducted the following conclusions were reached:

i. The both algorithm scored 98.9% in performance (accuracy, precision, recall and f1-score) of the classification technique onthe digital threats dataset.
ii. The Support Vector Machine is not suitable for training large-size dataset since it takes a lot of time to train.

6.2 Recommendation(s)

In this research, we described Random Forest and Support Vector Machine classification methods for detecting threats in a digital economy. In addition to the results shown in this research, the following recommendations were made:

i. The Support Vector Machine is too slow to train large chunks of dataset as in the case of our research dataset which can be a disadvantage, hence an alternative approach be employed to compensate for the drawback of the Support Vector Machine. The results of the Random Forest classification technique are tree-like structures that is easy to visualize compared with clusters of points.
ii. Since classification only indicates whether digital services are threat or not, therefore it will be interesting work to visualize the physical locations of digital threat services on a map.
iii. Implementation of the technique on different supervised machine learning algorithms to measure the performance.

REFERENCES

Aghaeikheirabady, M., Farshchi, S. M. R., & Shirazi, H. (2014, November). A new approach to malware detection by comparative analysis of data structures in a memory image. In *2014 International Congress on Technology, Communication and Knowledge (ICTCK) (pp. 1-4). IEEE.*

Alhanahnah, M., Lin, Q., Yan, Q., Zhang, N., & Chen, Z. (2018). Efficientsignature generation for classifying cross-architecture IoT malware. In *2018 IEEE conference on communications and network security (CNS) (pp. 1-9). IEEE.*

Bukht, Rumana & Heeks, Richards (2017). Defining, Conceptualising and Measuring the Digital Economy Centre for Development Informatics, University of Manchester, UK

Erdal Ozkaya, Milad Aslaner (2019). Hands-On Cybersecurity for Finance.Available *@subscription.packtpub.com*

Ekle*†,F.A, Agu†,N.M, Bakpo†,F.S, Udanor†,C.N and Eneh†,A.H.(2023)A machine learning model and application for heart disease prediction using prevalent risk factors in Nigeria.

Foley, J., Moradpoor, N., &Ochen, H. (2020). Employing a machine learning approach to detect combined internet of things attacks against two objective functions using a novel dataset. Security and Communication Networks, 2020, 1-17.

Hamad, A., Noor, F. and Arjun, B. C. (2019). Survey learning for Cybersecurity. *International Journal of Research in Electronics and Computer Engineering, 7(2). 2393-9028.*

Hansen, S. S., Larsen, T. M. T., Stevanovic, M., & Pedersen, J. M. (2016). An approach for detection and family classification of malware based on behavioral analysis. In *2016 International conference on computing, networking and communications (ICNC) (pp. 1-5). IEEE.*

He, K. and Kim, D. S. (2019). Malware detection with malware images using deep learning techniques. In 2019 *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 95-102). IEEE.*

Ieracitano, C., Adeel, A., Morabito, F. C. and Hussain, A. (2020). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing, 387, 51-62.

Ijaz, M., Durad, M. H., & Ismail, M. (2019). Static and dynamic malware analysis using machine learning. In *2019 16th International bhurban conference on applied sciences and technology (IBCAST) (pp. 687-691). IEEE.*

Kecman, Vojislav (2005) "Support vector machines–an introduction." *Support vector machines: theory and applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.1-47.*

Ren, Z., Wu, H., Ning, Q., Hussain, I.and Chen, B. (2020). End-to-end malware detection for android IoT devices using deep learning. Ad Hoc Networks, 101, 102098

Selvi, J., Rodríguez, R. J. and Soria-Olivas, E. (2019). Detection of algorithmically generated malicious domain names using masked N-grams. Expert Systems with Applications, 124, 156-163.

Takase, H., Kobayashi, R., Kato, M., & Ohmura, R. (2020). A prototypeimplementation and evaluationof the malware detection mechanism for IoT devices using the processor information. *International Journal of Information Security, 19, 71-81.*

International Journal of Computer Applications Technology and Research
Volume 13–Issue 04, 18 - 25, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1304.1003

Evaluation of various ML (Machine Learning) algorithms to detect attacks in Mobile Ad hoc Networks (MANETs) via AODV

N. Kanimozhi
Research Scholar,
Department of Computer Science, H.H The Rajah's College, Pudukkotai – 622 001.

Dr. S. Hari Ganesh
Assistant Professor,
Department of Computer Science, H.H The Rajah's College, Pudukkotai – 622 001.

Dr. B. Karthikeyan
Assistant Professor,
Department of Computer Science, Bishop Heber College, Trichy – 620 017.

Abstract: Infrastructure-less networks pose significant challenges in data security and time management, particularly within Mobile Ad-Hoc Networks (MANETs), where unstable mobile nodes can disrupt routing and affect Quality of Service (QoS) metrics. While numerous solutions exist for addressing these challenges, many of them introduce increased Overhead (OH) and Normalized Routing Load (NRL) to the MANET, which is unacceptable given the time-sensitive nature of communication sessions in MANETs. Exceeding processing or transmission times can lead to issues like Link Break (LB). The author proposes a solution to these challenges within the IDS-ATiC-AODV framework (Improved Data Security - Avoiding Time Complexity Ad-Hoc On-Demand Distance Vector). This IDS-ATiC AODV framework addresses five distinct qualitative and quantitative QoS issues using two primary algorithms. This study evaluates various Machine Learning algorithms from different clusters, leveraging the Infrastructure-Less Knowledge Measure Source Dataset (Infra-Less KMS Dataset) for analysis. This involves assessing the accuracy of each Machine Learning algorithm across ten different challenges using the Infra-Less KMS Dataset.

Keywords: MANET, Over Head, Normalized Routing Load, IDS-ATiC AODV, Infra-Less KMS Dataset

1. INTRODUCTION

The Mobile Node Network holds immense importance for the future due to its independence from existing infrastructure, making it the sole solution for military and emergency operations, particularly during natural disasters. Implementing MANETs and network routing presents significant challenges, both qualitatively and quantitatively. Addressing these challenges is crucial for MANETs to become indispensable networks. This research has already addressed Quality of Service (QoS) issues and introduced the IDS-ATiC-AODV algorithm group, capable of resolving various problems like.

- Black Hole Attach (BH)
- Grey Hole Attack (GH)
- Link Break (LB)
- End-to-End Time Delay (EETD)
- Data Theft (DT)(IDS)
- Data Change (DC)(IDS)

When a rule-based situation arises in the network during communication or at an opportune moment, IDS-ITiC promptly responds to mitigate the situation. Occasionally, this rule-based scenario provides pertinent information about ongoing attacks. However, there are instances where such attack-related information is unavailable. In such cases, the solution necessitates the execution of all relevant algorithms, leading to unnecessary processing. This redundancy often prolongs execution time, a scenario that commonly occurs.

MANET functions as a temporary network, establishing connections when a node (the source) seeks communication with its destination. For instance, if a node has a radio transmission range of 500 meters and its intermediate nodes travel at a speed of around 10 meters per second, the network availability from an intermediate node to the source would be less than 2 milliseconds. Thus, for the source node to maintain connectivity, it must complete its communication within this timeframe to avoid potential Link Break (LB) occurrences. Should communication exceed this window, the source node initiates a discovery process to establish a new route.

It is imperative that the solution algorithm does not escalate node overhead (OH) and load (NRL) along the transmission path by enlarging packet size, a responsibility entrusted to the solution algorithms. However, achieving this in practice proves challenging. Each sub-algorithm addressing specific issues necessitates the incorporation of certain parameters to monitor the situation effectively. These parameters are essential for the solution algorithm to identify occurring attacks and determine the appropriate algorithms to execute.

Given these constraints, reducing execution time is practically unfeasible. Therefore, this research aims to develop an automated solution. Specifically, the focus is on predicting attacks that occur at specific times. To achieve this, the author intends to integrate Machine Learning (ML) algorithms for attack prediction. Incorporating ML algorithms requires training them with a dataset, thus necessitating the acquisition of such data for this research.

The research has developed the Infra-Less KMS Dataset for evaluation purposes. Utilizing this dataset, the study will assess various categories of ML algorithms and subsequently recommend the most suitable algorithm for the IDS-ATiC-AODV routing protocol.

2. REVIEW OF LITERATURE

Elife Ozturk Kiyak et al. [1]: Our study introduces a machine learning model, termed high-level k-nearest neighbors (HLKNN), aimed at enhancing predictive analytics. This model enhances the classification capability of the KNN algorithm, commonly utilized across various machine learning domains. Given KNN's susceptibility to irrelevant data, its accuracy is often compromised by the quality of the training dataset. Experimental findings demonstrate that our developed model consistently outperforms KNN on well-known datasets, achieving average accuracy rates of 81.01% and 79.76%, respectively.

Xinhui Zhang et al. [2]: To achieve efficient online spatial data clustering, this research presents a DBSCAN extension algorithm integrating granular models tailored for online clustering. The proposed algorithm, structured into three layers via granular computing, constructs structural granules utilizing DBSCAN and GrC within the input space.

Thao-Trang Huynh-Cam et al. [3]: Experimental results indicate that the mean prediction accuracy rate of RF in 10-fold experiments was approximately 79.99%, DT achieved 74.59% by C5.0 algorithm and 80.00% by CART algorithm, and MLP reached 69.02%. CART surpasses C5.0, RF, and MLP algorithms.

Luca Scrucca [4]: This study introduces a novel approach to initializing the noise component in a Gaussian mixture model, offering an effective methodology for anomaly detection. The proposed approach involves an automatic procedure for selecting initial outlying observations to be used in the EM algorithm for Gaussian mixture models with a noise component. Specifically, noise initialization is based on comparing the contribution of each data point to the entropy of the Gaussian mixture with that arising from a uniform distribution over the hyper-rectangle enclosing the data.

Mohiuddin Ahmed et al. [5]: This paper addresses the popular k-means algorithm and its challenges regarding initialization and the handling of mixed feature types. Through critical analysis of existing literature and experimental assessments on benchmark datasets, the study reveals that each variant of the k-means algorithm is either application-specific or data-specific. Future research aims to develop a robust k-means algorithm capable of addressing both issues concurrently.

Nalindren Naicker et al. [6]: Results indicate that linear support vector machines outperform other methods in predicting student performance using student data. The algorithm suggests that parental education level does not influence student performance, whereas factors such as race, gender, and lunch have an impact. Future endeavors will explore ensemble methods of classical machine learning algorithms to enhance prediction accuracy.

Kumar S et al. [7]: Logistic regression proves to be a valuable tool in medical research for predicting binary outcomes and understanding the influence of predictor variables on patient health. By analyzing coefficients and odds ratios, clinicians can make informed decisions, personalize treatments, and advance medical knowledge, thereby facilitating evidence-based practice and revolutionizing patient care.

Jakub Horak et al. [8]: This study aims to develop bankruptcy prediction models and evaluate results obtained from classification methods, namely Support Vector Machines and artificial neural networks (multilayer perceptron artificial neural networks—MLP and radial basis function artificial neural networks—RBF).

3. MACHINE LEARNING ALGORITHMS

Machine learning algorithms come in a variety of forms, each intended to address a certain problem type and learn from data in a unique way. The following are a few major categories of machine learning algorithms:

1. Supervised Learning

Regression: Makes continuous result predictions.

Classification: Assigns a class or category to provided data.

2. Unsupervised Learning

Clustering is a type of unsupervised learning that puts comparable data points in a group. Reducing the amount of characteristics in the data without compromising its organisation is known as dimensionality reduction. Labelled and unlabelled data are combined for training in semi-supervised learning.

3. Reinforcement Learning:

This method of learning involves making mistakes, interacting with the environment, and getting feedback in the form of incentives or penalties.

4. Deep Learning:

Learns intricate patterns from vast volumes of data by utilising multi-layered neural networks.

5. Transfer Learning:

This method, which usually makes use of trained models, moves knowledge from one activity to another.

6. Ensemble Learning:

This technique, which uses Gradient Boosting or Random Forests, combines several models to increase performance.

7. Anomaly Detection:

Spots anomalies or odd trends in data.

These are only a few of the most common kinds of machine learning algorithms; there are many more particular algorithms with special traits and uses within each category.

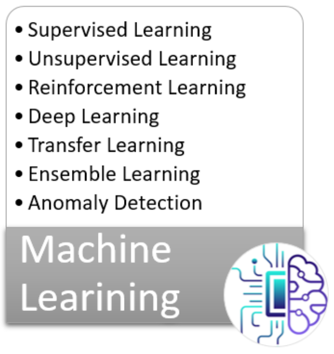


Figure 1: Different type of Machine Learning Algorithms

In this work, the author intends to utilize either supervised or unsupervised learning techniques.

Supervised learning involves training machines using meticulously labeled training data, allowing them to predict outputs based on this information. The labeled data indicates that certain input data is already associated with the correct output.

In supervised learning, the training data acts as a guide for the machines, instructing them on how to accurately predict outputs. This process entails providing both input and corresponding output data to the machine learning model. The objective of a supervised learning algorithm is to establish a mapping function that links the input variable (x) with the output variable (y).

In practical applications, supervised learning finds use in various domains such as risk assessment, image classification, fraud detection, and spam filtering.

During supervised learning, models are trained using labeled datasets, allowing them to familiarize themselves with different types of data. Following the completion of the training process, the model undergoes testing with a separate subset of the training data known as the test set, after which it generates predictions.

The operation of supervised learning can be comprehended through the following example and diagram:

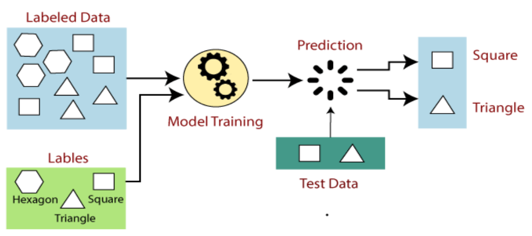


Figure 2: Working of Supervised Learning

Steps Involved in Supervised Learning:

1. Determine the type of training dataset required.
2. Collect or gather the labeled training data.
3. Divide the training dataset into subsets: training dataset, test dataset, and validation dataset.
4. Identify the input features within the training dataset, ensuring they provide sufficient information for accurate output prediction.
5. Select a suitable algorithm for the model, such as support vector machines, decision trees, etc.
6. Implement the algorithm on the training dataset, occasionally requiring validation sets for control parameters.
7. Assess the model's accuracy by evaluating its performance on the test set. A correct output prediction indicates model accuracy.

Types of Supervised Machine Learning Algorithms:

Supervised learning can be further categorized into two types of problems:

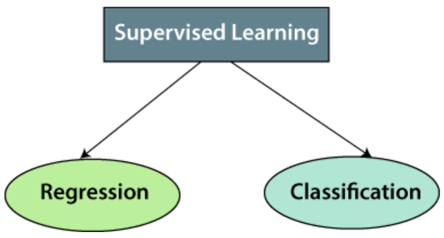


Figure 3: Major classification of Supervised Learning

1. Regression:

Regression algorithms are employed when a relationship exists between the input and output variables. They predict continuous variables, such as weather forecasting and market trends. Below are some prominent regression algorithms falling under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

2. Classification:

Classification algorithms come into play when the output variable is categorical, defining two classes like Yes-No, Male-Female, True-False, etc. They are utilized in applications such as spam filtering. Popular classification algorithms in supervised learning include:

- Random Forest
- Decision Trees
- Logistic Regression
- Support Vector Machine

3.1. Unsupervised Machine Learning

Unsupervised learning is a machine learning approach where models are not provided with a labeled training dataset. Instead, these models autonomously uncover hidden patterns and insights within the given data. It's akin to how the human brain learns new concepts without explicit instruction.

In essence, unsupervised learning entails training models with unlabeled datasets and empowering them to glean insights without external guidance. Unlike supervised learning, where input data is paired with corresponding output data, unsupervised learning operates solely on input data.

The primary objective of unsupervised learning is to discern the underlying structure of a dataset, group similar data points together, and represent the dataset in a more compact format.

The operation of unsupervised learning can be illustrated through the following diagram:

Clustering: Clustering involves grouping objects into clusters based on their similarities, where objects within a cluster share more similarities with each other than with objects in other clusters. Cluster analysis identifies commonalities among data objects and organizes them based on the presence or absence of these commonalities.

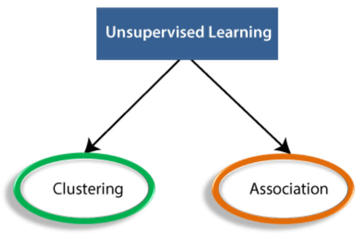


Figure 5: Major classification of Unsupervised Learning

Association: Association is an unsupervised learning technique used to discover relationships between variables within a large dataset. It identifies sets of items that frequently co-occur in the data, aiding in the formulation of effective marketing strategies. For instance, association rules can reveal patterns such as customers who purchase item X (e.g., bread) are also likely to buy item Y (e.g., butter/jam). Market Basket Analysis is a classic example of association rule usage.

Here is a compilation of some well-known unsupervised learning algorithms:

- K-means clustering
- K-nearest neighbors (KNN)
- Hierarchical clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Apriori algorithm
- Singular value decomposition (SVD)

4. DATASET

The upcoming dataset to be utilized in this study is the Infra-Less KMS Dataset, comprising approximately 31 attributes. These attributes are employed to analyze eight distinct factors encompassing both Qualitative and Quantitative Quality of Service (QoS) metrics. These factors include:

- DoS & DDoS
- Data Change
- Data Theft
- Block Hole
- EETD
- PDR
- (BU) Bandwidth Utilization
- NC(Network Congestion)
- UN(Unbelievable Node)

The following list presents the attributes contained within the Infra-Less KMS Dataset

- Com_ID: An exclusive communication identifier structured as SourceIP-DestinationIP-SourcePort-DestinationPort-TransportProtocol.
- IP-Source: The source IP address of the communication.
- Service-Source: The source port number.
- IP-Destination: The destination IP address.
- Service-Destination: The destination port number.
- Pro-ID: Identification number for the transport layer protocol (e.g., TCP = 6, UDP = 17).
- T_Instant: Timestamp indicating the capture time of the packet, in DD/MM/YYYY HH:MM:SS format.
- Com_Session: Total duration of the communication.
- Log_Fail: Number of failed login attempts.
- Log_Success: Binary indicator (1 if successfully logged in, 0 otherwise).
- Operation_Access_Control: Number of operations on access control files.
- SH_No_Of_Connection: Number of connections to the same host as the current connection within the past 2 seconds.
- Cur_Con_Ser_No.Of_Connection: Number of connections to the same service as the current connection within the past two seconds.
- Per_of_Connection_Diff_Hosts: Percentage of connections to different hosts.
- No_of_Same_Connection: Count of connections with the same destination host and service.
- Per_of_Connection_Diff_Hosts_Same_Ser: Percentage of connections to the same service originating from different hosts.
- No_Of_FW_Pkts: Total number of packets in the forward direction.
- No_Of_BW_Pkts: Total number of packets in the backward direction.
- Len_FW_Pkts: Total bytes in the forward direction across all packets in the communication.
- Len_BW_Pkts: Total bytes in the backward direction across all packets in the communication.
- Len_FW_Pkt_Max: Maximum packet length in bytes in the forward direction.
- Len_FW_Pkt_Min: Minimum packet length in bytes in the forward direction.
- Len_FW_Pkt_Mean: Mean packet length in bytes in the forward direction.
- Len_FW_Pkt_SD: Standard deviation of packet lengths in bytes in the forward direction.
- Len_BW_Pkt_Max: Maximum packet length in bytes in the backward direction.
- Len_BW_Pkt_Min: Minimum packet length in bytes in the backward direction.

- Len_BW_Pkt_Mean: Mean packet length in bytes in the backward direction.
- Len_BW_Pkt_SD: Standard deviation of packet lengths in bytes in the backward direction.
- Com_BPS: Bytes per second in the communication.
- Com_PktPS: Packets per second in the communication.
- One_Hop Neighbour: Updated when neighboring changes occur, structured as Source node-one hop node.

The table below provides details of the dataset including attack information.

Table 1: Display the number of rows allocated for each attack

Total Number of Rows	2,11,010.00
DoS (Denial of Service) & DDoS (Distributed Denial of Service)	32,021.00
DC (Data Change)	10,165.00
DT (Data Theft)	12,010.00
BH (Block Hole)	35,202.00
EETD (End to End Time Delay)	50,151.00
PDR (Packet Delivery Ratio)	42,435.00
BU (Bandwidth Utilization)	9,682.00
NC(Network Congestion)	9,669.00
UN(Unbelievable Node)	9,675.00

5. PROPOSED EVALUATION

The following algorithms are intended for use in this evaluation:-

1. Gaussian Naïve Byes (GNB)
2. Logistic Regression (LR)
3. K-Nearest Neighbours (KNN)
4. Linear Support Vector (LSV)
5. Decision Tree (DT)
6. Random Forest (RF)
7. Support vector Machines (SVM)
8. Gaussian Mixture Model (GMM)
9. K-Means (KM)
10. DBSCAN

Test dataset = 80% =1, 68, 810 rows

Train dataset =20% =42, 200 rows

5.1. Data Pre-processing

In this data pre-processing, the following tasks were performed: handling missing data, removing duplicates, identifying and handling outliers, encoding categorical labels, analyzing correlations, selecting relevant features, scaling the data, and splitting it into appropriate subsets.

Missing Data finding is done by the following code

```
total = train.shape[0]
missing_columns = [col for col in train.columns if train[col].isnull().sum() > 0]
for col in missing_columns:
    null_count = train[col].isnull().sum()
    per = (null_count/total) * 100
    print(f"{col}: {null_count} ({round(per, 3)}%)")
```

Duplicates is found following code.

```
train.duplicated().sum()
```

5.2. Feature selection

The subsequent code is utilized for feature selection in random forest classifiers.

```
rfc = RandomForestClassifier()
rfe = RFE(rfc, n_features_to_select=10)
rfe = rfe.fit(X_train, Y_train)
feature_map = [(i, v) for i, v in itertools.zip_longest(rfe.get_support(), X_train.columns)]
selected_features = [v for i, v in feature_map if i==True]
```

The provided code serves as a sample for training a Gaussian Naive Bayes (GNB) model on the BH (Black Hole) dataset

```
from sklearn.naive_bayes import GaussianNB
NB = GaussianNB()
NB.fit(X_train,y_train)
predictions = NB.predict(X_train)
print_stats(predictions, X_train, y_train, "Gaussian Naive Bayes on the train set")
```

6. RESULT AND DISCUSSION

Attack prediction percentage is listed in the following table. In this work around ten different attacks were predicted by the use of ten different Machine Learning algorithms. For the prediction it uses Infra-Less KMS Dataset. This dataset has around 2 lack records. Following table shows prediction accuracy with different attacks and different ML algorithms.

The preceding figure illustrates the prediction accuracy for attacks using Gaussian Naïve Bayes. According to these results, the algorithm's prediction accuracy falls within the range of 90% to 93%. While this range suggests promising accuracy, further validation against real-world scenarios is necessary to confirm its predictive capability.

Moreover, Gaussian Naïve Bayes exhibits notably low prediction accuracy of 90% for Data Theft and Packet Delivery Ratio. Conversely, it demonstrates favorable prediction accuracy for Data Change and End-to-End Time Delay, as indicated by these results.

The following figure shows the prediction accuracy with Logistic Regression. This algorithm gives high accuracy with Packet Delivery Ratio, Data Theft and Unbelievable Node, it gives 96% accuracy and it give low prediction accuracy 94% in Data change

Packet Delivery Ratio, Data Theft, and Unliveable Node exhibit higher accuracy rates. Conversely, Logistic Regression demonstrates lower prediction accuracy for Data Change.

Table 2: Prediction Accuracy between different ML Models

Attacks	**Accuracy (%)**									
	GNB	**LR**	**KNN**	**LSV**	**DT**	**RF**	**SVM**	**GMM**	**KM**	**DBSCAN**
PDR	90	96	99	97	99	100	92	81	69	29
EETD	93	95	99	95	65	98	78	74	70	52
DoS	92	95	99	96	52	80	85	77	69	41
DDoS	92	95	99	96	53	80	85	77	69	42
DC	93	94	35	94	48	65	71	67	65	48
DT	90	96	37	97	52	73	94	86	73	51
BH	92	95	52	96	99	99	83	76	69	51
BU	92	95	98	96	67	100	83	77	69	37
NC	91	95	99	95	100	100	80	73	67	39
UN	92	96	99	96	100	100	86	79	72	45

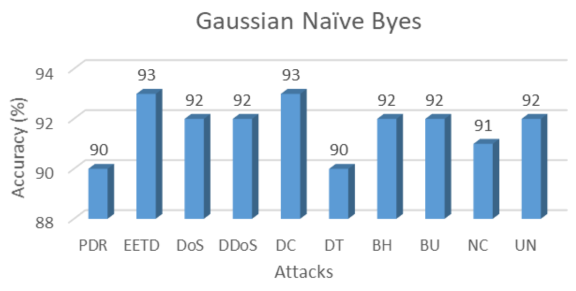


Figure 6: Prediction Accuracy between different attacks using Gaussian Naïve Byes

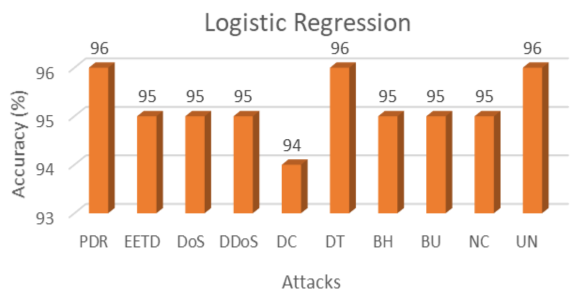


Figure 7: Prediction Accuracy between different attacks using Logistic Regression

The figure 8 above displays the prediction accuracy results for K-Nearest Neighbours, whereas the figure 9 below depicts the outcomes for Linear Support Vector. In comparison, Linear Support Vector yields moderate results.

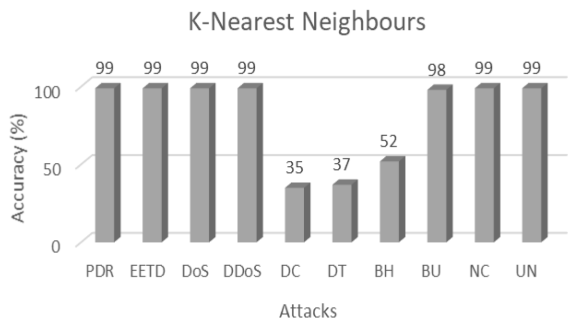


Figure 8: Prediction Accuracy between different attacks using K-Nearest Neighbors

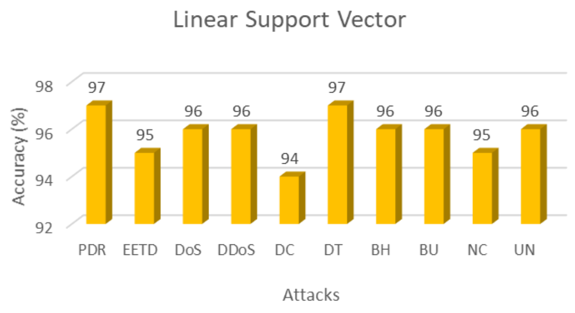


Figure 9: Prediction Accuracy between different attacks using Linear Support Vector

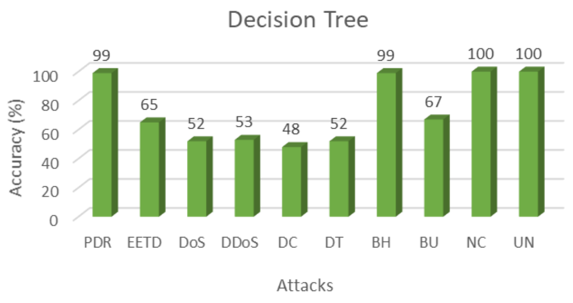


Figure 10: Prediction Accuracy between different attacks using Decision Tree

Decision Tree prediction accuracy is shown in the figure 10. Figure 11 shows Random Forest prediction accuracy. Random Forest gives moderate accuracy.

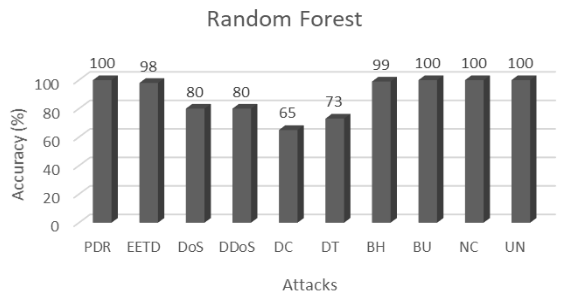


Figure 11: Prediction Accuracy between different attacks using Random Forest

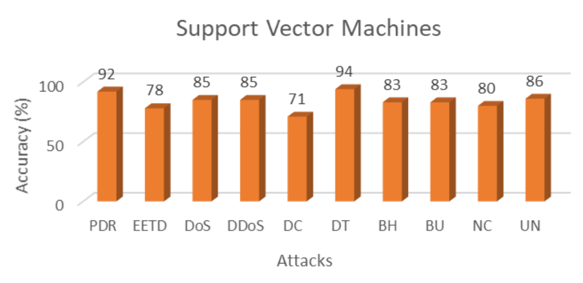


Figure 12: Prediction Accuracy between different attacks using Support Vector Machines

Figure 12 above indicates moderate prediction accuracy across all attacks, with rates ranging between 71% and 94%. This moderate level of accuracy, neither excessively high nor low, suggests its suitability for implementation in the IDS-ATiC-AODV routing protocol.

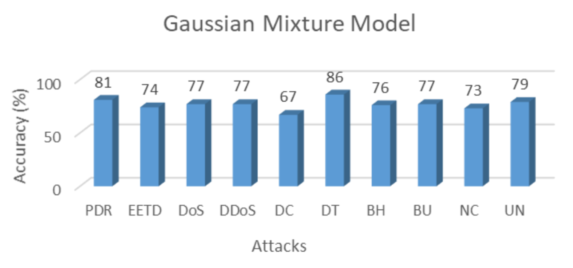


Figure 13: Prediction Accuracy between different attacks using Gaussian Mixture Model

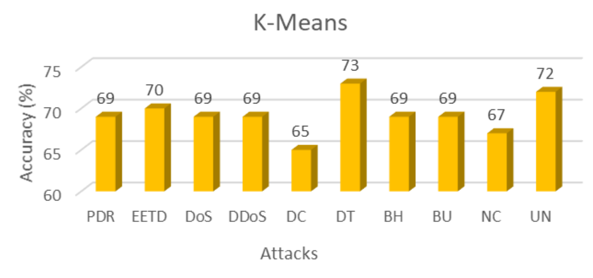


Figure 14: Prediction Accuracy between different attacks using K-Means

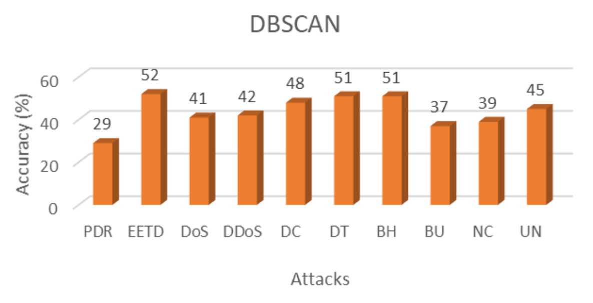


Figure 15: Prediction Accuracy between different attacks using DBSCAN

Figures 13, 14, and 15 depict the prediction accuracy for Gaussian Mixture Model, K-Means, and DBSCAN respectively. Notably, DBSCAN yields lower accuracy compared to Gaussian Mixture Model, which exhibits higher accuracy. However, the predictions from these three algorithms do not align with those from SVM.

7. CONCLUSION

Ten different ML algorithms accuracy prediction is evaluated with then different attacks by the use of Infra-Less KMS Dataset. This dataset has around 2 lack samples, in it 20% is used for training and remaining 80% of samples is used for testing. According to the evaluation some of the ML algorithms gives higher and lower prediction accuracy. But this research needs only on the moderate prediction accuracy giving ML Algorithms. So SVM is the moderate prediction accuracy provide ML algorithm. In the next work it will be using in the implementation with the IDS-ATiC-AODV routing protocol.

8. REFERENCES

[1] Elife Ozturk Kiyak, Bita Ghasemkhani and Derya Birant, "High-Level K-Nearest Neighbors (HLKNN): A Supervised Machine Learning Model for Classification Analysis",Electronics, 12, 3828, September 2023.

[2] Xinhui Zhang, Xun Shen, and Tinghui Ouyang, "Extension of DBSCAN in Online Clustering: An Approach Based on Three-Layer Granular Models", Appl. Sci., 12, 9402. September 2022.

[3] Thao-Trang Huynh-Cam, Long-Sheng Chen, and Huynh Le, "Using Decision Trees and Random Forest Algorithms to Predict and Determine Factors Contributing to First-Year University Students' Learning Performance", Algorithms, 14, 318. October 2021.

[4] Luca Scrucca, "Entropy-Based Anomaly Detection for Gaussian Mixture Modeling", Algorithms, 16, 195.April 2023.

[5] Mohiuddin Ahmed, Raihan Seraj and Syed Mohammed Shamsul Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation", Electronics, 9, 1295, August 2020.

[6] Nalindren Naicker, Timothy Adeliyi, and Jeanette Wing, "Linear Support Vector Machines for Prediction of Student Performance in School-Based Education", Hindawi Mathematical Problems in Engineering Volume 2020, Article ID 4761468,October 2020.

[7] Kumar S, Gota V., "Logistic regression in cancer research: A narrative review of the concept, analysis, and interpretation", Cancer Research, Statistics, and Treatment 2023;6:573-8,Dec-2023.

[8] Jakub Horak, Jaromir Vrbka and Petr Suler, "Support Vector Machine Methods and Artificial Neural Networks Used for the Development of Bankruptcy Prediction Models and their Comparison", Risk Financial Manag. 2020, 13, 60, March 2020.

[9] N. Kanimozhi, S. Hari Ganesh 2, B. Karthikeyan, "An Analysis of Machine Learning Solution for QoS and QoE in Network (Infrastructure Oriented and Less)", International Journal of Computer Sciences and Engineering, Vol.11, Issue 5, pp.41-59, May 2023, ISSN: 2347-2693 (Online), PP 41-59

[10] N. Kanimozhi, S. Hari Ganesh 2, B. Karthikeyan, "Performance Analysis of MANET Routing Protocols", International Journal of Computer Applications (0975 – 8887) Volume 185 – No. 50, December 2023, PP-44-50

[11] N. Kanimozhi, S. Hari Ganesh 2, B. Karthikeyan, "Irregularity Behaviour Detection - Ad-hoc On-Demand Distance Vector Routing Protocol (IBD - AODV): A Novel Method for Determining Unusual Behaviour in Mobile Ad-hoc Networks (MANET)", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11 Issue: 9, September 2023 pp 1098-1010.

[12] N. Kanimozhi, S. Hari Ganesh 2, B. Karthikeyan, "Minimizing End-To-End Time Delay in Mobile Ad-Hoc Network using Improved Grey Wolf Optimization Based Ad-Hoc On-demand Distance Vector Protocol (IGWO-AODV)", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11 Issue: 9, September 2023 pp 1111-1115.

[13] B.Karthikeyan, N. Kanimozhi and Dr.S.Hari Ganesh, “Analysis of Reactive AODV Routing Protocol for MANET”, IEEE Xplore (978-1-4799-2876-7), Oct 2014, pp. 264-267.

[14] B.Karthikeyan, N. Kanimozhi and Dr.S.Hari Ganesh, “Security and Time Complexity in AODV Routing Protocol”, International Journal of Applied Engineering Research (ISSN:0973-4562), Vol. 10, No.20,June 2015, pp.15542- 155546. – Scopus Indexed

.

[15] B. Karthikeyan, Dr.S.Hari Ganesh and N. Kanimozhi, “Encrypt - Security Improved Ad Hoc On Demand Distance Vector Routing Protocol (En-SIm AODV)”, ARPN Journal of Engineering and Applied Sciences (ISSN: 1819-6608), Vol. 11, No. 2, January 2016,pp. 1092-1096.

[16] B. Karthikeyan,Dr.S.Hari Ganesh and Dr. JG.R. Sathiaseelan, “ Optimal Time Bound Ad-Hoc On-demand Distance Vector Routing Protocol (OpTiB-AODV)”, International Journal of Computer Applications (ISSN:0975 – 8887), Vol. 140, No.6, April 2016,pp 7-11.

[17] B. Karthikeyan, Dr.S.Hari Ganesh, Dr. JG.R. Sathiaseelan and N. Kanimozhi , “High Level Security with Optimal Time Bound Ad-Hoc On-demand Distance Vector Routing Protocol (HiLeSec-OpTiB AODV)”,International Journal of Computer Science Engineering(E-ISSN: 2347-2693),Vol. 4, No. 4, April 2016, pp.156-164.