Modelling DDoS Attacks in IoT Networks Using Machine Learning

Pheeha Machaka[1,2(✉)], Olasupo Ajayi[2], Ferdinand Kahenga[2], Antoine Bagula[2], and Kyandoghere Kyamakya[3]

[1] University of South Africa, Johannesburg 1709, South Africa
machap@unisa.ac.za
[2] University of Western Cape, Cape Town 7535, South Africa
[3] Alpen-Adria-Universität Klagenfurt, 9020 Klagenfurt, Austria

Abstract. The Internet-of-Things (IoT) relies on the TCP protocol to transport data from a source to a destination. Making it vulnerable to DDoS using the TCP SYN attack on Cyber-Physical Systems (CPS). Thus, with a potential propagation to the different servers located in both fog and the cloud infrastructures of the CPS. This study compares the effectiveness of supervised, unsupervised, semi-supervised machine learning algorithms, as well as statistical models for detecting DDoS attacks in CPS-IoT.

The models considered are broadly grouped into three: (i) ML-based detection - Logistic Regression, K-Means, and Artificial Neural Networks with two variants based on traffic slicing. We also investigated the effectiveness of semi-supervised hybrid learning models, which used unsupervised K-Means to label the data, then fed the output to a supervised learning model for attack detection. (ii) Statistic-based detection - Exponentially Weighted Moving Average and Linear Discriminant Analysis. (Iii) Prediction 'algorithms - LGR, Kernel Ridge Regression and Support Vector Regression. Results of simulations showed that the hybrid model was able to achieve 100% accuracy with near zero false positives for all the ML models, while traffic slicing traffic helped improved detection time; the statistical models performed comparatively poorly, while the prediction models were able to achieve over 94% attack prediction accuracy.

Keywords: Anomaly Detection · Distributed Denial of Service · Internet of Things · Machine Learning · Regression Analysis

1 Introduction

The Internet of Things (IoT) provides a platform that allows objects to connect and communicate with one another using devices that can sense, identify and locate "things" in their surroundings, in order to better comprehend happenings in their environment. IoT devices are used for autonomous and intelligent tasks in residences, retail outlets office buildings, transportation [1], agriculture, healthcare [2], and manufacturing plants, among other places. The IoT market is growing at an exponential rate and is estimated

M. Masinde and A. Bagula (Eds.): AFRICATEK 2022, LNICST 503, pp. 161–175, 2023.
https://doi.org/10.1007/978-3-031-35883-8_11

to have grown to over 41 billion devices by 2027. Recently, the IoT has also expanded its reach beyond terrestrial networks by using drones [3] to complement the services delivered by semi-static IoT networks located on the ground [4–6]. The security of the complex network infrastructure resulting from the combination of terrestrial and airborne nodes, which use devices designed to operate in settings with limited resources (computing power, storage capacity, battery), is a challenging issue that requires incorporating security principles into different layers of the IoT protocol stack. For example, attacks such as Denial of Service (DoS) or Distributed DoS (DDoS) can be launched at the network, transport or application layers of the Internet stack, to easily compromise IoT devices [7] when such devices run routing protocols that use these layers for the transport of sensor readings, as illustrated on Table 1.

Table 1. IoT Specific Protocols

Protocol	Underlying protocol	Architecture	DDoS prone	Ref.
AMQP (RabbitMQ)	TCP	Publish/Subscribe	Yes	[8]
CoAP	UDP	Publish/Subscribe	No *	[9]
DDS	TCP	Publish/Subscribe	Yes	[10]
MQTT	TCP	Publish/Subscribe	Yes	[11]
XMPP	TCP	Both	Yes	[12]

Table 1 shows some IoT, and/or message telemetry specific protocols, including Message Queuing Telemetry Transport (MQTT), Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), and their corresponding underlying protocols. The table shows that protocols with publish/subscribe architecture rely on TCP protocol for data telemetry and are thus susceptible to DDoS based TCP SYN attacks. It is important to note that though Constrained Application Protocol (CoAP) does not run on TCP, it is still vulnerable to DDoS attacks such as UDP Flood.

Cyber Physical Systems IoT subsystem (CPS-IoT) [13], such as that shown in Fig. 1, rely on a mix of traditional IP networks and IoT specific protocols to move data from devices (physical and virtual sensors, actuators, edge devices and gateways) to / from the Cloud. An IoT specific protocol, such as MQTT or AMQP, is used for message telemetry between device(s) and the Fog infrastructure, as shown in Fig. 1; while an IP protocol, such as the Hypertext Transfer Protocol (HTTP), is used between the Fog and Cloud infrastructures. While both protocols (MQTT and HTTP) belong to different stacks, they are both guided by the TCP protocol in transporting data from source to destination. Hence, DDoS attacks such as the TCP SYN can be plausible tools that attackers use to mislead the operation of CPS and potentially cause critical damages.

Having shown through Fig. 1 and Table 1 that CPS-IoT data telemetry protocols mostly run on TCP/IP - HTTP (TCP port 80 or 8080) and MQTT (TCP port 1883 or 8883) [11] or AMQP (TCP port 5671 or 5672) [8] - we now focus on modelling DDoS attacks on the underlying TCP/IP network layer in the rest of this paper.

Due to the Internet's phenomenal development over the last few decades, attackers now have access to a growing number of vulnerable devices and often use the IoT subsystem of CPS (where these devices are located) to launch vicious attacks that can adversely affect the CPS as a whole. For instance, an attacker may use a large number of these susceptible devices to initiate an attack on a server located in a Fog close to the devices or in a Cloud infrastructure located far away. These attacks often have various modes of intensity, with attacks that are perpetrated with low intensities, often able to evade detection by current detection techniques.

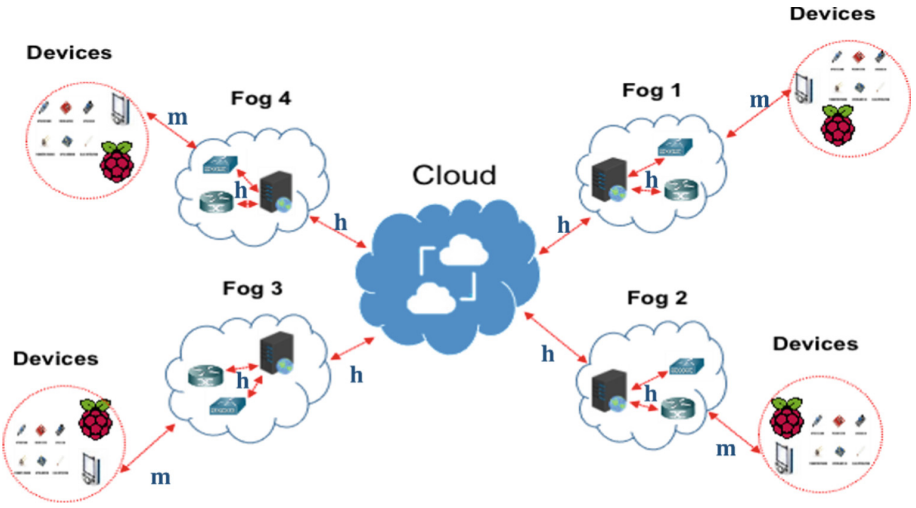

Fig. 1. A Generic CPS-IoT Subsystem

Through this research we explore the application of machine learning (ML) models, including classification and prediction, to model DDoS attacks, specifically SYN attacks in IP networks, such as those upon which CPS-IoT subsystems are built. A potential use case of our research is in sensor virtualization in CPS-IoT systems. In this use case, virtual sensors are in the Fog or Cloud infrastructure to enhance real sensors with the capability of differentiating and classifying incoming traffic into genuine or bogus traffic in real-time. This discerning ability is a key requirement for the efficient operation of next generation CPS, where security would be of paramount importance. The selection of the most efficient algorithms for the classification of the sensor data traffic and the prediction of future attacks on the CPS-IoT are other key requirements for ensuring the safe operation of CPS infrastructures. However, these processes are beyond the scope of this work.

The specific contributions of this work include:

- Comparison of the efficiency of supervised, unsupervised, semi-supervised ML models and statistical models in modelling DDoS attacks, in a bid to distinguish between safe and adversarial network traffic.

- The development of a semi-supervised learning model, capable of auto-labelling traffic and using the labelled traffic to accurately identify malicious traffic. This is achieved by hybridizing supervised and unsupervised machine learning models.
- Determining the impact, if any, of splitting network traffic into window sizes versus using the entire traffic stream in detecting malicious attacks.
- Exploring the effectiveness of regression models in predicting potential DDoS attacks, in a bid to move the safety of IP networks from reactive to proactive.

The rest of the paper is structured as follows, related literature is reviewed in Sect. 2, while our research methodology is presented in Sect. 3. Section 4 gives details of our implementation process and obtained results, while Sect. 5 concludes the paper and gives insights into potential future research directions.

2 Literature Review

The first DDoS assault on the public Internet happened in August 1999 [14]. In February 2000, a year after the initial event, several commercial websites, including Yahoo, CNN, and eBay, saw their first DDoS attacks. A high number of requests overloaded these websites, forcing their services to go offline which resulted in considerable financial losses. The July 4 2009 cyber-attacks are well-known examples of DDoS attack, where prominent government, news media, and financial websites were targeted in a series of cyber-attacks across South Korea and the United States [15]. Researchers have worked on techniques to combat DDoS attack even in the context of IoT, for example authors in [16] surveyed IoT related security challenges and potential solutions for attacks such as DoS. The three key technologies that form the basis of the majority of today's detection techniques are machine learning (ML), information theory, and statistical models [17]. Artificial Neural Networks (ANN), support vector machine (SVM), and other ML techniques in cybersecurity are helpful for decision making analysis [18]. The paragraphs that follow highlight some of the related work in application of ML to DDoS attack detection.

In order to detect DDoS attacks, the authors in [19] proposed combining feature selection with an ANN MLP (multilayer perceptron) model. This strategy was used to choose the best features during the training phase, and they created a feedback system to reconstruct the detector when significant detection faults were detected dynamically. With a 98% accuracy rate, the proposed methodology proved effective.

Chaudhary et al. [20] also suggested a ML technique for detecting DDoS assaults that involved filtering crucial network packet parameters such as packet size and interval size. SVM, Random Forest, Decision Tree, and Logistic Regression were used and Random Forest surpassed the other models with a DDoS attack detection accuracy of 99.17%. In [21], the authors used flow features of network traffic, such as packet size, packet interval, protocol, bandwidth, and destination IP, to construct a model to detect DDoS attacks. They used SVM, K-Nearest Neighbour (KNN), Random Forest, Decision Tree, and ANN in their models. The results of the experiment showed that Random Forest and ANN have 99% accuracy in detecting malicious traffic.

For detecting DDoS attacks in Software Defined Networks (SDN), [22] employed SVM, KNN, ANN, and Naive Bayes. Initially, the authors specified twelve features, but

the algorithms chose a subset of these features based on threshold values. The algorithms analysed flow traffic data and detected DDoS with 98.3% accuracy.

The accurate and timely detection of DDoS attacks remains a priority for researchers in the field of cybersecurity, however, attackers keep modifying and developing new attacks in order to evade detection techniques. In this research study we distinguish between normal and DDoS attack network traffic and compare the performance of supervised, unsupervised, and semi-supervised machine learning techniques.

Additionally, the efficacy of two approaches for forecasting possible DDoS attacks was investigated. In the section that follows, we will provide a detailed account of the methodological approach followed in this study.

3 Methodology

Figure 2 gives an overview of the proposed system. The important components are data pre-processing, supervised learning, semi-supervised learning, unsupervised learning and prediction. Each of these components described as follows:

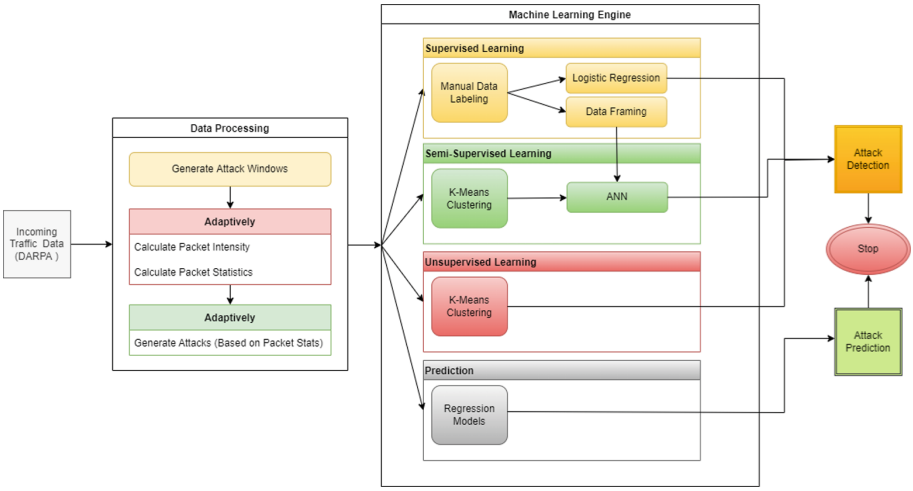

Fig. 2. Proposed System Architecture

3.1 Data Pre-processing and Labelling

For this work, we used the DAPRA IDS evaluation dataset [23], which was prepared by the MIT Lincoln Laboratory under DARPA and AFRL sponsorship. We used this dataset because we had earlier inferred that IoT systems have an underlying IP network upon which they run, hence still vulnerable to classic IP attacks such as DDoS. The tcpdump format was used, wherein all network activities, including the whole payload of each packet, were recorded and supplied for assessment.

We processed the raw dataset by writing a Python script to count the number of network packets that arrived at a given host per 10 s interval. We used this as each 10 s block as zero (0), corresponding to no DDoS attack [24]. We then introduced malicious attacks to the dataset by manually increasing the number of packets arriving in randomly selected intervals. We labelled these as one (1), implying DDoS attack.

3.2 Supervised Learning

This component is labelled "Supervised Learning" in Fig. 2 and it involved applying supervised ML on manually labelled data. Supervised learning is a class of machine learning (ML) wherein an ML model is trained using pre-labelled data, which serve as "examples" for the ML model. Once the model has been trained, it can then be exposed to new (test) data for classification or prediction. In our system, we considered Logistic Regression (LGR) and Artificial Neural Network (ANN) models.

Data Framing.
Data framing was done for ANN only and three variants were considered. In the first, data framing was not considered, and this served as the base line; while in the second, the dataset was split into "frames" of size 12, corresponding to 120 s of traffic flow (at 10 s interval). In the third, the standard deviation of values in the frame was calculated and appended to the frame, thus increasing the frame size to 13. The data frames were then fed to the ANN model. The data framing process is summarized with the pseudocode in Algorithm 1.

Algorithm 1: Data Framing Algorithm

- Divide the entire dataset into data blocks of 120 seconds.
- For each 120 second data block in the dataset:

1. Create a 3 by 4 data frame as follows:
   - Set t = 0
   - For row = 1 to 4

   a) *col1 = Packet □ Count(t); t+ = 10*
   b) *col2 = Packet □ Count(t); t+ = 10*
   c) *col3 = Packet □ Count(t); t+ = 10*
2. Calculate the stand deviation (σ) for the data block. //for option 1 only

For each data frame, the standard deviation (σ) is calculated. This standard deviation is used to further verify the probability that a malicious attack has occurred. Within a data frame, if the data points are far from the mean, then the deviation of values within the frame would be higher, which implies that an attack occurred in that data frame. The opposite holds true for data points that are closer as their deviation from the mean would be smaller. This can be interpreted as an absence of attack(s). Finally, in instances where all 12 entries in a data frame are high (full DDoS attacks), the standard deviation value from the mean would be small. To distinguish between this full attack situation and a

safe situation, a threshold value is used. If the calculated σ is greater than this threshold value, then the frame is classified as being under attack.

Machine Learning Models.
As mentioned above, both LGR and ANN were considered in this work. For LGR, an 80:20 split was used for training and test data, using the One-over-rest (OvR) training scheme and linear memory BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm. In this work, we used it to model DDoS attacks in IP networks and we considered a 3 layered ANN architecture. At the input layer we had 12 or 13 nodes (σ included), the hidden layer had 6 nodes, while the output layer had 1 node. The data frames obtained in the previous subsection were fed in, with the corresponding standard deviation value used as the 13th node. The ReLU (Rectified Learning Unit) activation was used for the input and hidden layers, while Sigmoid activation was used at the output layer. The processes involved in our ANN supervised learning component are depicted in Fig. 3.

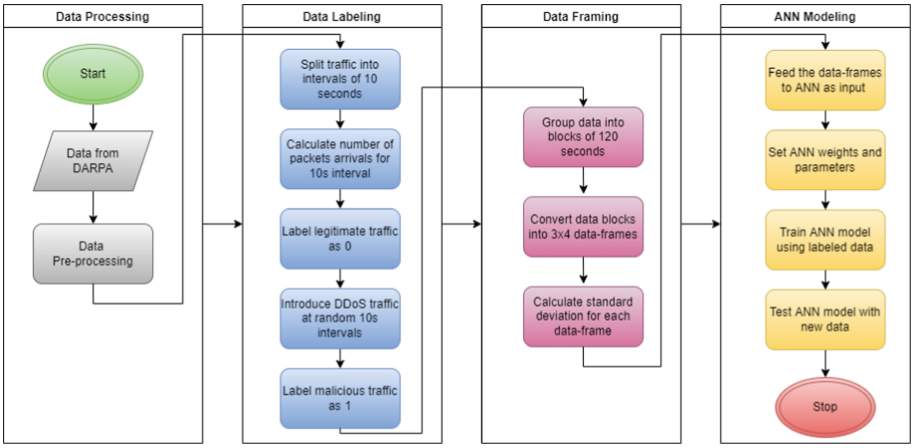

Fig. 3. ANN Supervised Learning Process

3.3 Unsupervised Learning

As an alternative to manually labelling the dataset, we considered the K-means clustering technique for automatic labelling. K-Means is a centroid based clustering algorithm that determines cluster membership based on the proximity of data points to a centre point (centroid) [25]. In IP networks security, millions of packets often traverse the network per unit time and need to be classified (labelled) as either legitimate or malicious traffic. Manually doing this would be slow and laborious in such cases, hence the use of an automatic classifier is desirable, in our case K-Means. In our work, traffic flow falls into one of two categories (legitimate or malicious), thus, k value is set to 2.

3.4 Semi-supervised Learning

Our semi-supervised learning component, which is labelled "Semi-Supervised Learning" in Fig. 2, is like the supervised learning described earlier. The major difference is that rather than using manually labelled data as input to the ML models, we fed the output of the unsupervised learning (K-Means clustering) into the models. In essence, K-Means is used to automatically label (classify) the data, which is then used to train the supervised model. Thus, creating a semi-supervised model. The output of this model is then compared to the two other models (supervised and unsupervised).

3.5 Statistical Models

For completeness, we performed data classification using a classic statistical model – the Exponentially Weighted Moving Average (EWMA) [26]. By placing more emphasis on recent data points than on older ones, EWMA can detect anomalies in observed data quickly. In applying EWMA, we sought to detect anomalies within blocks of data. An anomaly would be a disruption from the "norm" (normal traffic flow). Such anomalies are considered as attacks. We set a window size of 12, corresponding to 120 s, then measured the deviations from the average traffic count in each window. The steps for calculating EWMA are well documented in literature, however, a concise summary of our application is given in Algorithm 2.

Algorithm 2: EWMA Algorithm

- Set a window size of 12 (data blocks of 120 seconds).
- Set mean, std, thresholdUp, thresholdDw to 0
- For each window:
  1. mean += calculate the moving average.
  2. std += calculate the moving standard deviation.
  3. *thresholdUp* = *mean* + *std*
  4. *thresholdDw* = *mean* - *std*
  5. Slide the window by 1 (10 seconds)
- For each data point (d) in the dataset:
  1. If d > *thresholdUp* or d < *thresholdDw* : attack = True.
  2. Else attack = False.

Linear Discriminant Analysis (LDA) is a statistical model used for feature reduction and distinguishing between data entries in a dataset. In this work we are only concerned with its application in data classification abilities, specifically binary classification of data traffic into normal or attack. The steps of the 2-dimensional (binary) LDA classification are well documented in literature, having initially being proposed in the early 1930s by Fisher [26]. For brevity the steps are repeated in this work but refer interested readers to the work of Fisher for details.

3.6 Prediction

Having successfully classified and distinguished between legitimate and malicious attacks, the next logical step might be to predict the possible occurrence of such attacks. This would help the network administrator put preventive measures in place to mitigate them, essentially changing the defence strategy from reactive to proactive. This is highlighted in green in Fig. 2. Three regression models were considered in this work for prediction, the Logistic Regression (LGR), Kernel Ridge Regression (KRR) and Support Vector Regression (SVR).

4 Implementation

For this work, implementation was carried out on Google Colab, with a Python 3 Google Compute module, configured with 12 GB of RAM, 2.3 GHz 2 Core Intel Xeon CPU and GPU hardware accelerators. Keras and Sci-Kit learn were used for machine learning; Smote was used for data balancing; Pandas, NumPy were used for data manipulation, while matplotlib was used for data visualization. Finally, an 80:20 split was used for training and testing data for the supervised learning algorithms.

4.1 Metrics

Six metrics were used to compare the performance of the models considered, these are false positive, false negative, average execution time, accuracy, coefficient if determination (R^2), and Root Mean Square Error (RMSE). The first 4 are specific to classification models, while the last 3 (accuracy inclusive) are for the regression models.

4.2 Supervised Learning

Table 2 summarizes the results of the supervised learning models. From the table, the pure ANN model resulted in the highest accuracy, followed by ANN + Data framing + SD. Logistic Regression (LGR) also had high accuracy value but resulted in the highest number of false negatives, meaning that LGR wrongly classified more legitimate traffic as attacks. Conversely, data framing adversely affected the false positive rate, resulting in about 2% of bogus traffic (DDoS attacks) being misclassified as safe. Among the three ANN models considered, the variant without data framing only slightly edged out the variant with data framing and standard deviation (SD) at 99.414% vs. 99.405%. The impact of Data framing + SD is also evident here as the combination resulted in the lowest false negative of all the models compared.

W.r.t execution time, Fig. 4 shows that the pure ANN was the slowest of all four models, taking over 2 min to classify traffic flow. This would be unacceptable in real-time environments, where high-speed data analysis and classification are paramount. In contrast, the variants of ANN based on data framing were significantly faster than both LGR and the pure ANN at just 11s vs 51s and 130s respectively. This shows that breaking traffic into data frames or "windows" and processing them accordingly can be significantly beneficial with regards to processing time.

4.3 Unsupervised Learning

Running the K-Means classifier with K = 2, resulted in an accuracy of 96.76%, with zero false positives.

Table 2. Anomaly Detection Using Supervised Learning Models

Model	Accuracy	False positives	False negatives
LGR	99.192	0	1.6215
ANN	99.414	0	0.6695
ANN + Dataframing	98.842	2.1805	0.1295
ANN + Dataframing + SD	99.405	0.9565	0.0965

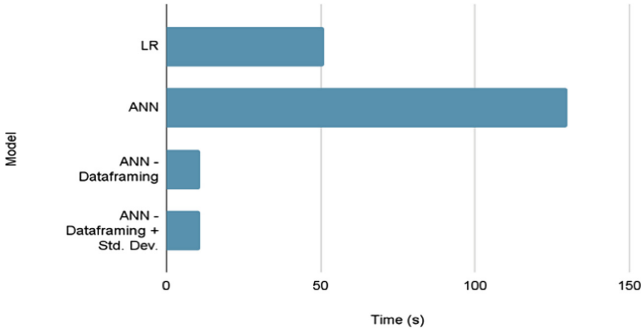

Fig. 4. Average Execution Time

4.4 Semi-supervised Learning

The labelled outputs from K-Means (unsupervised learning) were used as input to the supervised learning models, in essence creating a form of semi-supervised model. Table 3 shows the performance of this hybrid combination. From the table the incorporation of the K-Means classifier resulted in a significant boost in the performance of all the models. Both LGR and the pure ANN resulted in perfect accuracies, zero false positives and zero false negatives.

Similarly, the accuracies of both variants of ANN with data framing increased from 98.842% to 99.64% and 99.41 to 99.69% respectively. Of important note is the reduction in false positive and false negative values of ANN + Data framing and ANN + Data framing + SD. Respectively. For the former, the false positive dropped from about 2.18% to just 0.73%, while the false negative value dropped to 0.07. For ANN + Data framing + SD, the false positive value dropped to 0.67%. The overall improvements in the results on Table 2 compared to Table 3 shows the efficacy of our proposed hybrid (semi-supervised) model in detecting malicious attacks. However, the fact that both variants of ANN did

not yield 100% accuracy cannot be ignored. A possible explanation for this is that the dataset was not split into data frames of equal sizes, hence some data frames (especially those at the tail end of the traffic flow) contained less data i.e., less than the window size (12 data points).

Table 3. Anomaly Detection Using Semi-Supervised Models

Model	Accuracy	False positive	False negative
K-Means + LGR	100	0	0
K-Means + ANN	100	0	0
K-Means + ANN + Data framing	99.64	0.73	0.07
K-Means + ANN + Data framing + SD	99.69	0.67	0.01

4.5 Statistical Models

Table 4 summaries the results of the statistical methods used for detecting malicious (abnormal / attack) traffic. For comparison purposes, we also included the result of the pure Logistic Regression model (LGR).

Table 4. Anomaly Detection Using Statistical Models

Model	Accuracy	False positives	False negatives
EWMA	71.299	12.102	57.268
LDA	99.837	7.797	37.870
LGR	99.192	0	1.6215

Compared to LGR, both EWMA and LDA performed poorly w.r.t False Negatives and False Positives. The false positive and negative values in EWMA are understandably high because the model uses simple moving standard deviation and mean of observed samples to determine differentiate attacks. This means that for every subsequent traffic window (120 s interval), EWMA would compare the mean and standard deviation of that window with its preceding window. If the difference is much, EWMA flags that window as being attacked. To elaborate, if we assume that little or no data traffic arrive during the first 120 s, EWMA establishes a base line with this first window size using the mean and standard deviation (SD). If during the next few seconds, significant number of legitimate traffic arrive, EWMA calculates the mean and SD of this new block. It then compares the new mean and SD with the base line. The new values would be higher than the base line and EWMA would flag this new window as malicious because of the higher traffic count. The reverse is the case with the false negatives.

Being probabilistic (based on Bayes' theorem), LDA expected performs better than EWMA in most of the metrics. However, like EWMA, LDA also struggled with distinguishing between high volume legitimate traffic and malicious traffic. This problem becomes more pronounced when low traffic windows(s) is/are followed by window(s) with slightly higher traffic counts. In such instances, the succeeding window(s) would be classified as malicious even if there are not.

4.6 Prediction

As stated earlier, three prediction models were considered, and their results are summarized on Table 5. Of the three models compared, LGR performed the best, with a prediction accuracy of 98.6%. It was closely followed by KRR at approx. 98%. SVR was the least accurate of the lot at 94.64%. For R2, values closer to 1 are desirable, and depicts the "closeness" of predicted values to the actual values. For the three models, the same trend is observed with R2 scores, as LGR led with a score of approx. 0.94, followed by KRR at 0.91. SVR scored 0.76, implying that its prediction curve differed greatly from the actual curve. Finally, for RMSE, values closer to 0 are desirable as they indicate lower prediction errors. Once again, LGR was the least error prone as it had the lowest RMSE values, followed by KRR with a score of 0.1439. However, both models were less error prone than SVR with a RMSE value of 0.2314. We can thus conclude that LGR is the best predictor, while KRR is a close alternative. With such high RMSE value, SVR is a less than ideal predictor in our use case.

Table 5. Comparison of Results of the Prediction Models

Metric	KRR	LGR	SVR
Accuracy	97.93%	98.60%	94.64%
R^2	0.9054	0.9361	0.7555
RMSE	0.1439	0.1183	0.2314

Overall, these results show that LGR and KRR are better prediction models than SVR for our use case. With attack prediction accuracies of approximately 98% for both LGR and KRR models, it can be inferred that regression models can be used to predict potential DDoS attacks in IoT networks. For both LGR and KRR, the inaccurate predictions were in instances where they assumed that attacks would occur when none occurred. These wrong predictions or false alarms, though leading to unnecessary deployment of defensive mechanisms, are preferable to the reverse case. In the reverse case, as observed with SVR, the model gives a false sense of security by predicting that no attack would occur, when imminent threats abound. We therefore consider the wrong predictions of KRR and LGR as "erring on the side of caution".

5 Conclusion

In this study, the accuracy and timeliness of supervised, unsupervised, and semi-supervised machine learning techniques for detecting Distributed Denial of Service (DDoS) attacks in Cyber Physical-Internet of Things Systems (CPS-IoT) were explored. CPS-IoT systems often rely on two well-known protocols for data transmission, namely HTTP and MQTT, both of which are built upon TCP/IP, hence vulnerable to TCP/IP targeted attacks. DDoS attacks are common to TCP/IP, thus pose a potential threat to the security, dependability and safety of CPS-IoT systems. In this work, five machine learning models (ML) and two statistical models were considered for modelling DDoS attacks in IoT networks (TCP/IP-based). These are Logistic Regression (LGR), Artificial Neural Networks (ANN), K-Means, Kernel Ridge Regression (KRR), Support Vector Regression (SVR), Exponentially Weighted Moving Average (EWMA) and the Linear Discriminant Analysis (LDA).

In distinguishing between normal traffic and bogus (attack) traffic, two supervised ML classifiers were used - LGR and ANN (and two slight variations of ANN based on slicing). LGR gave a classification accuracy of 99.19%, a false positive rate of 1.62%, and an average detection latency of 51 s from the initiation of the attack. The ANN model, on the other hand, had better accuracy at 99.41% and lower false negative value of 0.67%, but was extremely slow at 130 s. We introduced slicing, and split the traffic into fixed windows sizes, before applying ANN. This slicing improved the false negative values and significantly cut down the detection time to just 11 s. We then considered the K-Means unsupervised ML model, which resulted in 96.76% classification accuracy. Finally, we developed semi-supervised ML models by combining the K-Means with the ANN and LGR. These combinations resulted in a classification (detection) accuracy of 100% with near zero false positives across all models. Compared to the ML models, the statistical models performed poorly w.r.t false positive and negatives.

We further examined the use of regression models to support network administrators in transiting from reactive to proactive network management approach. LGR, KRR, and SVR were investigated for their abilities to correctly predict attacks before they occur. LGR gave the best prediction accuracy at 98.6%, followed by KRR at 97.9%, while SVR had the worst performance at 94.64%. The R^2 values for the LGR and KRR were 0.94 and 0.91 respectively, representing closeness to actual values, while their RMSE values were respectively 0.12 and 0.14. SVR was significantly off the mark for these metrics. In essence, LGR and KRR are both capable for predicting imminent threats, with LGR being slightly better.

This work only considered traffic counts, time and status in determining DDoS attack. In future works, other features such as source and destination IP addresses or ports can be considered. Similarly, only the possible future attack times were considered, potential future research could consider the target machine or subnet. Finally, in CPS-IoT systems, attack detection and prediction using network topology graphs could be another avenue for future research work.

References

1. Ajayi, O.O., Bagula, A.B., Maluleke, H.C., Odun-Ayo, I.A.: Transport inequalities and the adoption of intelligent transportation systems in Africa: a research landscape. Sustainability **13**(22), 12891 (2021)
2. Bagula, A., Mandava, M., Bagula, H.: A framework for healthcare support in the rural and low income areas of the developing world. J. Netw. Comput. Appl. **120**, 17–29 (2018)
3. Ismail, A., Bagula, B.A., Tuyishimire, E.: Internet-of-Things in motion: a UAV coalition model for remote sensing in smart cities. Sensors **18**(7), 2184 (2018)
4. Ma, K., Bagula, A., Nyirenda, C., Ajayi, O.: An IoT-based Fog computing model. Sensors **19**(12), 2783 (2019)
5. Zennaro, M., Bagula, A.: Design of a flexible and robust gateway to collect sensor data in intermittent power environments. Int. J. Sens. Netw. **8**(3–4), 172–181 (2010)
6. Bagula, A.B.: Hybrid traffic engineering: the least path interference algorithm. In: Proceedings of the SAICT 2004, ACM International Conference Proceedings Series, pp. 89–96 (2004). ISBN: 1-58113-982-9
7. Ahmad, R., Alsmadi, I.: Machine learning approaches to IoT security: a systematic literature review. Int. Things **14**, 100365 (2021)
8. AMQP: CloudAMQP. https://www.cloudamqp.com/docs/amqp.html
9. Pardo-Castellote, G.: Omg data-distribution service: architectural overview. In: Proceedings of IEEE Military Communications Conference (MILCOM), pp. 200–206 (2003)
10. Anonymous "MQTT FAQ." https://mqtt.org/faq/
11. Millard, P., Saint-Andre, P., Meijer, R.: "No title," XEP-0060: Publish-Subscribe, XMPP Standards Foundation
12. Bagula, A., Ajayi, O., Maluleke, H.: Cyber physical systems dependability using CPS-IOT monitoring. Sensors **21**(8), 2761 (2021)
13. Garber, L.: Denial-of-service attacks rip the Internet. Computer **33**(04), 12–17 (2000)
14. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun. Surv. Tutorials **15**(4), 2046–2069 (2013)
15. Khan, F.I., Hameed, S.: Understanding security requirements and challenges in internet of things (IoTs): a review. arXiv preprint arXiv:1808.10529
16. Singh, K., Singh, P., Kumar, K.: Application layer HTTP-GET flood DDoS attacks: research landscape and challenges. Comput. Secur. **65**, 344–372 (2017)
17. Hosseini, S., Azizi, M.: The hybrid technique for DDoS detection with supervised learning algorithms. Comput. Netw. **158**, 35–45 (2019)
18. Wang, M., Lu, Y., Qin, J.: A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Comput. Secur. **88**, 101645 (2020)
19. Chaudhary, P., Gupta, B.B.: Ddos detection framework in resource constrained internet of things domain. In: Proceedings of IEEE Global Conference on Consumer Electronics (GCCE), pp. 675–678 (2019)
20. Wehbi, K., Hong, L., Al-salah, T., Bhutta, A.A.: A survey on machine learning based detection on DDoS attacks for IoT systems. In: Proceedings of the IEEE Southeastcon, pp. 1–6 (2019)
21. Polat, H., Polat, O., Cetin, A.: Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. MDPI Sustain. **12**(3), 1035 (2020)
22. Lichman, M.: DARPA intrusion detection evaluation dataset. DARPA Intrusion Detection Evaluation Dataset—MIT Lincoln Laboratory (2000)
23. Machaka, P., Bagula, A.: Statistical properties and modelling of DDoS attacks. In: Vinh, P.C., Rakib, A. (eds.) Context-Aware Systems and Applications, and Nature of Computation and Communication. ICCASA ICTCC 2020 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 343. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67101-3_4

24. Morissette, L., Chartier, S.: The k-means clustering technique: general considerations and implementation in Mathematica. Tutorials Quant. Methods Psychol. **9**(1), 15–24 (2013)
25. Roberts, S.W.: Control chart tests based on geometric moving averages. Technometrics **1**(3), 239–250 (1959)
26. Theodoridis, S.: Classification: a tour of the classics. In: Theodoridis, S., Ed. Machine Learning, pp. 275–325. Academic Press, London (2015)