Andrushchak, I., & Kosheliuk, V. A. (2024). Integration of machine learning algorithms for intrusion detection in IoT networks. *Actual Issues of Modern Science. European Scientific e-Journal, 32*, __-__. Ostrava: Tuculart Edition, European Institute for Innovation Development.

DOI: 10.47451/inn2024-07-01

The paper is published in Crossref, ICI Copernicus, BASE, Zenodo, OpenAIRE, LORY, Academic Resource Index ResearchBib, J-Gate, ISI International Scientific Indexing, ADL, JournalsPedia, Scilit, EBSCO, Mendeley, and WebArchive databases.

Igor Andrushchak, Doctor of Technical Sciences, Professor, Department of Software Engineering, Lutsk National Technical University. Lutsk, Ukraine.
ORCID: 0000-0002-8751-4420, Scopus: 54882165900
Viktor A. Kosheliuk, Candidate of Engineering Sciences (Ph.D.), Associate Professor, Department of Computer Science, Lutsk National Technical University. Lutsk, Ukraine.
ORCID: 0000-0002-4136-5087

Integration of machine learning algorithms for intrusion detection in IoT networks

Abstract: The Internet of Things (IoT) is a powerful technology that is transforming many aspects of our lives, from how we connect and work to how we receive healthcare and manage the economy. IoT holds promise for enhancing life across diverse settings, from urban environments to educational institutions, through task automation, productivity enhancement, and stress reduction. As new threats and vulnerabilities emerge, the old ways of securing IoT devices are no longer sufficient. The future of secure IoT systems relies on machine learning and deep learning that are optimized for efficiency. To ensure robust security in constantly evolving next-generation IoT systems, we need to harness the power of artificial intelligence, particularly machine learning and deep learning solutions. To achieve this vision of constantly adapting security for next-generation IoT, the authors must create entirely new methods that guarantee the highest levels of security within the entire IoT infrastructure. The study subject is detection systems for intrusions into IoT infrastructure and compromised IoT devices based on machine learning algorithms. The study object is a machine learning model that will be able to detect anomalies in the behavior of an IoT network and identify patterns that indicate normal behavior and deviations that may signal an intrusion. The study aims to enhance the security of IoT networks by developing effective and efficient intrusion detection systems using machine learning techniques. During the study, such scientific methods as data collection and preprocessing, algorithm selection and development, model training and evaluation, experimentation and analysis, scalability and efficiency testing were used. The authors used the works of such scientists and researchers as A. Géron, N. Sengupta, R. Vinayakumar, S. Sarwar, Wang Meng. The study investigates security mechanism for understanding attacker behavior in the realm of the IoT. This could be a significant step forward in fortifying IoT security. This approach to securing IoT devices relies on machine learning to analyze the data traffic these devices produce during communication. Additionally, this paper proposes incorporating machine learning methods to enhance honeypot operation by integrating them into the lambda function's design. Machine learning is becoming increasingly popular across many fields because it often performs better than traditional rule-based approaches. While the idea of fully automated cyber security detection and analysis using machine learning is appealing, it is essential to carefully assess how well machine learning works in this area. The authors offer an analysis tailored for security professionals, focusing on utilizing machine learning techniques in developing a honeypot designed for detecting intrusions.

Keywords: Internet of Things, honeypot, lambda function, MQTT, machine learning.

Introduction

The world is becoming more and more “smart” as the Internet of Things (IoT) turns ordinary objects into portals to the digital world. The network of IoT physical objects equipped with sensors, software and other technologies is capable of collecting, processing and exchanging data over the Internet. IoT is currently growing rapidly. As these networks expand and become more complex, so does the range of their capabilities. IoT is already being used in a wide variety of areas: from household appliances that automatically order products to sophisticated industrial equipment that optimises production (*Abdallah Wasan, 2023*).

The evolution of IoT technologies has made them a force that spans many industries and is no longer new. Sensors, automation, networks, data collection, and analytics, like miniature information processing devices, are all components of this evolution. A combination of technologies: sensors, automation, networks, data collection and analytics, like miniature information processing devices, is the basis of this evolution. Incorporating these components into various objects results in the creation of intelligent vehicles, drones, instruments, and other machines. Security and privacy concerns can arise due to the IoT devices and cyber-physical systems (CPS) variety. This problem is becoming more acute due to the growing dependence on IoT in areas such as online shopping, banking, education, and doing business (*Liang &amp; Kim, 2021*; *Sengupta, 2019*; *Salau et al., 2022*).

By using communication protocols, the Internet of Things is a network that allows objects to be connected. Connecting IoT nodes to the Internet has been made possible through multiple protocols, such as TCP/IP using the MQTT queue message transport protocol, Modbus TCP, and LoRaWAN technologies (*Kasongo, 2021*). Communication protocols incorporate some safeguards to protect against various types of threats, such as data theft, brute-force attacks, port scanning, DoS and DDoS attacks, MITM, R2L, and probing attacks. There are two types of IoT attacks: user root (U2R) attacks and operating system attacks (*Sarwar et al., 2022*).

Detecting cyber threats can be achieved by using honeypots and honeynets, which can provide valuable information about attackers’ actions. A honeypot is a device that can attack and potentially steal information (*Fan at. al., 2018*). The network becomes a Honeynet when two or more Honeypots are deployed. IPS formation can be accomplished by integrating honeypots with firewalls and IDS, obtaining complete information about attackers, studying their actions, developing strategies to enhance system security, and preventing similar attacks in the future.

The study subject is detection systems for intrusions into IoT infrastructure and compromised IoT devices based on machine learning algorithms.

The study object is a machine learning model that will be able to detect anomalies in the behavior of an IoT network and identify patterns that indicate normal behavior and deviations that may signal an intrusion.

The study aims to enhance the security of IoT networks by developing effective and efficient intrusion detection systems using machine learning techniques. By achieving these aims, the research seeks to provide a significant advancement in the security mechanisms available for IoT networks, leveraging the power of machine learning to protect against the increasing number of cyber threats targeting these environments.

Based on the set purpose, the following tasks are set:

- gathering diverse IoT network traffic data, including normal and attack scenarios;
- training and evaluating machine learning models;
- comparing different machine learning algorithms and their combinations;
- evaluating the system in real-world IoT environments.

The presented work proposes using machine learning (ML) and deep learning (DL) methods to build the lambda function through the honeypot intrusion detection method. Machine learning, which can analyse large amounts of data and detect patterns, looks like a promising method for improving intrusion detection systems in IoT environments. Adaptive solutions based on data are offered that can identify anomalous patterns and behaviour in real-time. Machine learning algorithms have demonstrated particularly effective results in identifying potential threats owing to the significant amount of data generated by IoT devices (*Spitzner, 2020*). The quality and significance of the datasets used to train these algorithms determine their effectiveness.

By combining these scientific methods, our study proposes the integration of machine learning algorithms to detect intrusions in IoT networks and contribute to the development of robust security solutions.

The authors used the works of such scientists and researchers as A. Géron, N. Sengupta, R. Vinayakumar, S. Sarwar, Wang Meng.

Study problem

The communication protocol defines a standard way for two or more entities to establish meaningful interactions, enabling valid, legitimate, and expected behaviour by all involved. In the case of IoT, application layer protocols can define not only how IoT devices exchange information, but also how the devices are managed as part of the IoT platform. Given that protocols define the fundamentals of expected behaviour, we can identify anomalies (that may be part of malicious activity) through dynamic protocol analysis, including payload (content) processing, context, and common patterns. data), analyse it, and perform specific processes. analysis can be done. For this purpose, various approaches can be implemented, such as traffic analysis, honeypot techniques, and protocol analysis of all systems involved in communication.

Honeypot technology

Honeypot technology is a mechanism that intercepts an attacker's activities by simulating a real system but placing it in a protected environment. An attacker can reach a honeypot once it is recognised as a real system or device. Honeypots are implemented in a protected and monitored manner so that while the attacker's information is being recorded, the attacker's activities do not harm the system. The main concept of honeypot technology is that a communication protocol running on service (software) acts as a decoy system or trap for intruders. The level of interaction can be defined as the range of possibilities that a honeypot offers to an attacker. Generally, there are three types of honeypots in terms of interaction level (*Wang, 2017*).

High Interaction Honeypot (HIH) is essentially a real system that uses standard protocol implementations. Its main feature is that, as a real system, it allows full interaction with the attacker. However, security concerns must be considered as exploits may occur in the real world. Therefore, HIH involves the use of monitoring and network control systems that protect the environment during the attacker's activities. IoT platforms can be implemented using XMPP/MQTT/REST HIH honeypots.

Low Interaction Honeypot (LIH) detects attackers by employing software emulation to mimic the characteristics of specific operating systems, applications, network services, or protocols on the host operating system. This approach offers several advantages. Firstly, attackers operate within a simulated environment, reducing the risk associated with real exploitation scenarios. Secondly, the emulation provides greater control over the attacker's actions, allowing for more detailed monitoring and analysis of their activities. However, there are also drawbacks to this approach. While LIH emulates services or steps within a protocol, it may not fully replicate the design or functionality of the targeted applications or protocols. This limitation can affect the effectiveness of data collection and interaction with the attacker. Examples of LIH implementations include Dionaea, Honeyd, NetBait, and Kippo. LIH can also be utilized for emulating IoT devices, and it can interact with both real and emulated XMPP and MQTT services.

Medium Interactive Honey Pot (MIH). Honeypots provide attackers with more interaction opportunities than low-interaction honeypots but have fewer features than high interaction solutions, known as medium-interaction honeypots. They may expect certain activities and be designed to provide predetermined responses beyond what a low-interaction honeypot would offer. MIH combines features of LIH and HIH, but they can be more complex in design and implementation. The proposed IoT honeypot prototype is related to MIH.

IoT application protocols

Communication protocols establish a consistent method for two or more entities to engage in meaningful interaction, ensuring proper and expected behaviour from all parties involved. Because there are no universally defined standards for all IoT components, the technologies employed by IoT platforms vary in their features. Essentially, any technology that meets the connectivity criteria can be utilised. Presently, various companies have developed their unique IoT architectures.

No matter the type of wireless technology employed, the data from end-devices can be made accessible on the internet through two methods (*Wang, 2017*): transmitting information to a specialised web service or Application Programming Interfaces (API) that can be accessed via the internet; utilising cloud-based platforms.

These web services, APIs, or cloud platforms serve as the database for storing and processing data, act as an intermediary node between devices and end-users, and provide APIs that enable end-users to remotely monitor and control the devices.

Numerous application protocols have been identified as suitable for IoT communication. Some of these include MQTT, XMPP, AMQP, CoAP, UPnP, JMS, HTTP REST, and DDS. Each protocol possesses distinct characteristics and can be applied in various scenarios. Furthermore, they can collaborate by being implemented in different segments of an IoT system. Several surveys have compared these protocols, assessing their suitability for IoT based on factors such as reliability, security, and energy consumption.

Security issues can be considered from three perspectives: protocol flaws, implementation issues, and integration vulnerabilities. Is crucial for ensuring robust security in IoT systems. When implementing protocols with IoT platforms, it is essential to leverage the security mechanisms inherent in the protocols themselves. Table 1 (*Wang, 2017*) provides a summary of the security mechanisms employed by the communication protocols mentioned earlier.

During protocol implementation, such as building and installing a server, security issues can arise due to development-related factors like bugs, weaknesses, or inadequate validations. These issues can introduce vulnerabilities into the overall implementation, potentially leading to security breaches. These vulnerabilities may eventually be documented in the Common Vulnerabilities and Exposures (CVE) database. The CVE database maintains a comprehensive list of known vulnerabilities for various software products, including operating systems, libraries, frameworks, and both open-source and closed-source implementations.

Indeed, various databases maintain a shared list of CVE identifiers, such as CVE Details (*Vulnerability List…, 2023*) and the NVD NIST databases (*Merenda et al., 2020*). These repositories provide comprehensive information about vulnerabilities, including details about the affected vendor, product, time of discovery, vulnerable versions, vulnerability type, description, and more. By searching for CVEs associated with IoT-related protocols, developers can identify potential exploitation vectors for IoT applications or platforms that utilise these protocols. It is crucial for developers to regularly update their systems and implement solutions to mitigate the risk of exploitation from known vulnerabilities. This proactive approach helps enhance the security posture of IoT deployments and reduces the likelihood of successful cyber-attacks.

As IoT platforms integrate various technologies and protocols, they become susceptible to security attacks. These vulnerabilities often emerge during the integration of different IoT application protocols, signalling potential challenges in IoT security. To address these issues, stakeholders can implement various approaches to collect, analyse, and identify threat patterns targeting IoT platforms. This proactive stance helps mitigate potential security risks and fosters a more secure environment for IoT deployments.

Machine Learning techniques

Recent academic studies have demonstrated the effectiveness of AI technologies, specifically Machine Learning (ML), in monitoring cybersecurity (*Géron A., 2019*). ML's ability to create a model capable of learning the statistical patterns within different datasets enables it to make predictions without the necessity of explicitly coding a set of rules.

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that enables computers to learn without explicit programming. It entails crafting a predictive algorithm specific to each problem at hand. These algorithms learn from data to recognise patterns and trends, thereby constructing a model for prediction or classification. Deep Learning, a subset of Machine Learning, employs multiple layers to extract increasingly complex features from raw input. The term "deep" in "deep learning" refers to the depth of layers used in data transformation. Many Deep Learning algorithms rely on Artificial Neural Networks (ANN) (*Zhang et al., 2021*).

Constructing Machine Learning (ML) methods can be computationally demanding when dealing with intricate datasets, necessitating significant memory and time resources.

Consequently, ML techniques must undergo careful optimisation to function effectively in resource-constrained environments resembling the Internet of Things (IoT). The premise is that feature reduction can lower the training cost of ML algorithms using a given dataset. Subsequently, it introduces an optimisation approach capable of generating a lightweight ML technique that consumes minimal memory and execution time while accurately distinguishing between attacks and regular traffic on IoT networks (*Moustafa et al., 2019*).

Intrusion detection is predominantly a binary classification task with one main goal: detecting or classifying whether a traffic sample is part of an attack. But in today's world of specialization, with more data to analyze and more complex devices in the infrastructure, attacks need to be classified in more detail for proper countermeasures and future fixes and workarounds. Binary classification alone is not enough to properly deal with detected threats; a more granular classification is required. Therefore, classifying groups of attacks or specific attacks is the task of a multi-category problem.

Machine learning systems can be grouped based on the level and manner of supervision they receive during training, with three main classifications: supervised learning, unsupervised learning, reinforcement learning. The supervised classifiers that underwent training and evaluation belonged to five separate categories (*Banaamah & Ahmad, 2022*; *Shone et al., 2018*; *Tuna et al., 2022*): decision trees (DT), random forest (RF), convolutional neural networks (CNN), recurrent neural networks (RNN) and long short-term memory (LSTM).

The achievement in enhancing the resource efficiency of the LGBM technique encourages further exploration of additional AI technologies, particularly those based on Deep Neural Networks (DNN). Recent studies have highlighted the effectiveness of DNN in intrusion detection, surpassing many traditional ML models in cybersecurity monitoring. However, a drawback of DNN-based approaches is their demand for substantial resources to construct a model capable of achieving superior detection accuracy with a multidimensional feature set.

This poses a challenge in training scenarios like edge machine learning, where smart devices can process data locally using machine and deep learning algorithms (e.g., federated learning). Furthermore, IoT devices, in contrast to mainstream IT devices, have constrained computing resources (processing and storage) to ensure maximum data output with minimal energy consumption, while also being cost-effective. Consequently, DNN-based security solutions tailored for mainstream IT devices cannot be directly applied for security monitoring in environments with limited computing resources.

This necessity arises from the constraints of IoT resources, such as memory and processing power, coupled with the resource-intensive nature of existing AI-driven cybersecurity approaches for handling complex multidimensional data. Therefore, the outcomes of this research can offer valuable insights to security professionals and industries on implementing secure, resilient, and efficient AI solutions in resource-constrained settings. Moreover, other cybersecurity researchers can leverage the techniques introduced in this thesis to enhance current AI security solutions within IoT network environments.

Literature

The evolution of technology has brought forth new cyber/physical attack vectors that pose significant challenges in identification and assessment. Integrating IoT-enabling technologies with air-gapped legacy cyber/physical systems, particularly in expansive and intricate environments like critical infrastructures, has rendered the assessing risk task within these domains exceptionally challenging. Even with the utilisation of well-established risk assessment methodologies, evaluating the risk in any one of these domains is inherently daunting.

Vinayakumar et al. (*Vinayakumar et al., 2019*) delve into the exploration of Deep Neural Networks (DNNs) for constructing an adaptable and efficient intrusion detection model. This model aims to detect and categorise unplanned and unpredictable cyber-attacks within a network, leveraging various freely available cyber community malware datasets. Given the dynamic nature of malware attacks, the study's objective is to identify the most effective algorithms for detecting cyber threats. Vinayakumar et al. propose the Scaled-hybrid_IDS model, which employs hybrid DNNs for detecting malware within the network. This model is designed to monitor cyber-attacks at both the host level and network traffic in real-time environments.

To identify an effective machine learning algorithm for intrusion detection or cyberattacks within IoT-based smart city applications, introduced a machine learning selection framework utilising a bijective soft set approach and its associated algorithm (*Shafiq et al., 2022*). The Bot-IoT dataset was employed for evaluating this framework. Among the algorithms assessed, including NB, BayesNet, C4.5, RF, and Random Tree, the NB machine learning algorithm emerged as the preferred choice for anomaly and intrusion detection of IoT device attacks in smart cities. This algorithm demonstrated superior performance in terms of accuracy and the time required to build the model compared to the other algorithms evaluated.

Sequeiros et al. (*Sequeiros et al., 2020*) provide an overview of related research concerning attack and threat modelling for IoT systems and cloud mobile applications. On the other hand, the authors introduce IotCom, an approach aimed at uncovering concealed threats. Specifically, the researchers investigated multi-app coordination threats capable of initiating infinity activation loops or chain coordination events that may result in race conditions and physical wear of a device. Through their platform, they conducted static analysis of multiple IoT applications and identified numerous instances of safety violations.

In their study, Chen et al. (*2020*) conducted a review focusing on IoT application cyber-attacks within smart city environments, specifically addressing detection and classification using deep learning algorithms. The authors explored various deep learning models including deep belief networks, Boltzmann machines, restricted Boltzmann machines, CNNs, recurrent ANNs, and generative adversarial networks for attack detection and classification within smart cities. Furthermore, they presented several deep learning-based cyber-attack detection models tailored for IoT applications within smart city contexts.

One of the myriad challenges confronting the Internet of Things (IoT), which integrates a diverse array of objects into networks to facilitate sophisticated and intelligent applications, is safeguarding user privacy and thwarting various attacks, including spoofing, denial of service (DoS), jamming, and eavesdropping. The author (*Sangra, 2023*) examines the vulnerabilities present in IoT systems and explores potential strategies to bolster the security of IoT networks utilising machine learning techniques such as supervised learning, unsupervised learning, and reinforcement learning (RL). The analysis of data privacy delves into ML-based approaches for tasks such as authenticating IoT devices, regulating access to these devices, securely offloading data, and identifying viruses.

Materials and methods of research

This study focuses on deploying honeypots in AWS EC2 and utilising machine learning techniques to create a lambda function. The aim is to entice potential cyber criminals to engage with these deployed honeypots, thereby gathering a substantial amount of data for analysis. Because low-interaction honeypots are easily replicated, modifying default service banners and settings will help to make them more realistic. The main objectives are to monitor harmful intruders' behaviours, assess their origins, accumulate different attack strategies, and collect malware samples and payloads.

When crafting an IoT solution, it is crucial to grasp the potential threats it may face and implement defence in depth by incorporating multiple security measures. These measures should cover identification, protection, detection, and response to threats. Designing the solution with security in mind from the outset is crucial because comprehending how an unauthorised individual could potentially compromise the system enables the implementation of appropriate mitigations.

The AWS IoT security baseline (AISB) outlines a collection of security controls that establish a minimum foundation for customers to construct secure IoT solutions on the AWS platform. In the AISB solution architecture, an IoT device transmits data to AWS IoT Core. This data from the edge device is then forwarded to AWS for tasks such as processing, storage, analytics, and visualisation. In addition to telemetry data, AWS IoT Device Defender allows IoT and IoT devices to report security events directly to AWS. This event information is merged with cloud-based events to pinpoint security misconfigurations, identify anomalies in device behaviour, and alert personnel to respond promptly to security events. The principle of operation solution architecture AISB is presented in the appendix (*Figure 1*).

While deploying honeypots (decoy devices mimicking real IoT systems seems straightforward – just setting up boxes with simulated IoT software – this approach has limitations. It might only capture a narrow range of attack data. The longer an attacker interacts with a honeypot (a decoy device mimicking an IoT system), the more valuable information we gain about their goals and methods. As attackers become more invested in a seemingly real device, the honeypot needs more complexity to maintain the illusion and gather richer intel.

Given the intricate interaction an IoT device has with its environment, an IoT honeypot needs to be structured in a manner that enables intelligent adaptation to diverse types of traffic. The success of this ongoing battle is gauged by the quantity of valuable insights gained relative to the engineering effort invested. The authors' purpose is to construct a meticulously designed system comprising a range of honeypot devices operating in coordination with a vetting and analysis infrastructure.

Iot Core Frame. The AWS IoT core services are comprised of five different services that are responsible for maintaining the needs of all IoT devices, connecting to the AWS cloud, managing devices, updating over the air (OTA), and safeguarding the IoT devices. Within this framework, the TLS communication protocol encrypts all communication. Rules facilitate interaction between IoT devices and AWS services.

AWS IoT Core provides security through policies and X.509 certificates, along with support for MQTT over TLS/SSL. An AWS IoT Core policy is a JSON document encompassing one or more statements. These statements consist of three types: effect, which determines whether the action is permitted or denied; action, specifying the action permitted or denied by the policy; and resource, identifying the resource or resources on which the action is permitted. This policy lets devices connect to AWS IoT Core if their client ID (device identifier) is the same as their thing name (a unique name assigned in AWS). Additionally, devices can publish data to any topic that starts with their thing name. Instead of relying on usernames and passwords, AWS IoT Core uses a more secure method for devices to publish data. Devices need to identify themselves with special certificates called X.509 certificates. These certificates are unique to each device and are created by AWS IoT Core after the device is registered (becomes a "thing").

MQTT protocol. At its core, AWS IoT Core relies on a messaging protocol called MQTT to communicate with devices. This protocol acts like a middleman, separating the devices that send data (publishers) from the ones that receive it (consumers). Devices simply publish their data, and the MQTT broker efficiently routes and delivers the messages to the interested parties. This approach keeps things flexible and scalable.

Rules are analysed, and actions are executed based on the MQTT topic stream. Topics serve to identify AWS IoT messages. A message broker is employed to assign topic names and topic filters, routing messages sent via MQTT and HTTP to the Hypertext Transfer Protocol Secure message URL.

Devices publish data using organised topic names that act like addresses. To receive specific data, services subscribe to matching "topic filters". These filters act like sieves, sorting messages based on their topic names and delivering them only to the relevant subscribed services. Large-scale IoT deployments, like those managing farms with thousands of devices, can get complex. To simplify this, AWS offers a "shadow service". This service creates a virtual representation (shadow) of each real device in the cloud. In the context of your smart livestock system, each animal's sensor would have a corresponding shadow in the AWS cloud. This approach enables each of the utilised devices to be accessed and managed distinctly by various services. These modifications are enacted either through the MQTT protocol or via HTTP using the device shadow REST API.

Lambda Frame. AWS lambda functions are short pieces of code that run on demand. Unlike traditional applications, they do not require constant server maintenance. They take input, process it, and produce an output. These functions can be triggered by various events within or outside of AWS, making them highly versatile. One key benefit is automatic scaling. Lambda functions can handle a surge in traffic without you needing to manually adjust server capacity. A Lambda, in contrast to an EC2 instance, is designed to run for a single purpose and is only meant to run for a short while. Lambda functions require essentially no platform maintenance and scale immediately to hundreds of instances.

Machine Learning Frame. Infrequently accessed data is stored in S3 Glacier (Serverless), designed for long-term data archiving. Unlike S3 Buckets, it is not readily accessible, and intended solely for archived content. If needed, this data can be unarchived and restored to S3. Subsequently, it can be effectively utilized within the machine learning framework, where the data trains machine learning algorithms for regression or classification predictions. The development, training, and deployment of the interface models were handled by Amazon SageMaker.

In our study, we implemented the most common machine learning algorithms (*Wang, 2017*) to create a lambda function in the AWS IoT security framework to detect multi-vector cyberattacks in the IoT:

- decision tree (DT);
- K-Nearest neighbour (KNN);
- random forest (RF);
- support vector machine (SVM);
- extreme gradient boosting (XGBoost).

Architecture for ML inference is presented in the appendix (*Figure 2*).

Results

Cloud computing provides on-demand access to computing power and storage, using virtual machines that can be easily scaled up or down based on your needs. Cloud computing removes limitations on processing power and storage. Applications can access the immense computing resources of cloud data centres, eliminating the need for expensive on-site hardware. The architecture of honeypots is entirely constructed within the AWS cloud environment.

This research proposes an architecture built entirely on serverless services offered by AWS such as AWS lambda, Amazon S3, Amazon SNS, Amazon API Gateway, Amazon DynamoDB, etc. This allows us to create data pipelines that can efficiently handle the large amount of data coming from IoT devices.

To run the experiment, we set up a virtual server on Amazon's cloud following T-Pot's recommendations. This server had the processing power and memory of a t3.xlarge instance type. The virtual machines used for the experiment all ran the Debian 12 operating system on a special virtualisation technology called HVM (Hardware Virtual Machine). The instance had a decent amount of processing power with 2 virtual CPUs (vCPUs), 8 Gigabytes of memory (RAM), and a high-speed network connection capable of handling up to 5 Gigabits of data per second.

The evaluation metrics listed and defined in Table 2 are used to assess the performance of feature extraction algorithms and machine learning models. TP, FP, TN, and FN denote the counts of True Positives, False Positives, True Negatives, and False Negatives, respectively.

Experimental results are presented in the appendix (*Tables 3*; *Table 4*; *Table 5*; *Table 6*). The total comparisons of the different MLA efficiencies of detecting attacks on the main IoT Core communication protocols such as MQTT, HTTPS, MQTT over WSS, and Hybrid are presented in the appendix (*Tables 3*; *Table 4*; *Table 5*; *Table 6*). The Hybrid connection method means that devices connect to AWS IoT Core for management, but receive data through other means, such as Amazon Kinesis Data Streams, Amazon MSK, Amazon SQS, or Amazon API Gateway.

Discussion

As technology advances and creates more data, cyber-attacks will likely become both more common and more complex. The importance of cybersecurity is growing rapidly on the list of priorities for governments around the world. To keep our information safe online, it is important to be able to spot cyber-attacks. These detection systems can find unusual activity and warn people about threats so they can take action to protect themselves.

This research explores how machine learning can be used to build honeypots, being tools for detecting cyberattacks. Machine learning and deep learning are becoming popular tools in many areas, including cybersecurity. However, it is significant to figure out which types of these techniques work best for different cybersecurity problems. This thesis centers on exploring powerful machine learning methods for building cutting-edge systems that can detect cyberattacks.

For IoT security to be truly effective, it needs to rely on machine learning or deep learning models that leverage data attributes. To make smart choices, the system needs a powerful learning algorithm that considers both its knowledge of IoT security and the specific task it is designed for.

The more sophisticated IoT and cloud computing become, the more crucial cloud platforms become for managing them effectively. The AWS cloud platform from Amazon provides a wide array of Infrastructure as a Service (IaaS) components like Platform as a Service (PaaS) offerings. For powerful and efficient storage of data collected from Internet of Things (IoT) devices, you can combine Amazon's IoT Core service with their Rules engine and DynamoDB storage.

Conclusion

This research has investigated how effective honeypots are at detecting cyberattacks in an Internet of Things (IoT) setting. We achieved this by simulating real-world IoT devices and deploying honeypots within that simulated environment. Honeypots are decoy computer systems designed with vulnerabilities to attract attackers. This clever strategy diverts their attention away from real, critical systems. By mimicking real systems and recording attacker activity, honeypots act as a secret weapon. They allow us to gather valuable information about what attackers are after, how they operate, and the tactics they use.

This article describes an improvement to honeypots designed for IoT devices using AWS's IoT Core platform. This improvement leverages machine learning to better defend against attackers who use similar techniques. We tested our honeypot with a modified Lambda function on AWS. The honeypot successfully fooled attacker tools designed to sniff out honeypots and even tricked attackers into uploading malicious software. How well machine learning can spot multi-pronged attacks on IoT systems depends heavily on the quality of the data used to train and test these algorithms. We examined the feasibility of detecting attacks on Internet of Things (IoT) infrastructure by focusing on the most commonly used connection methods in IoT, including MQTT, HTTPS, MQTT over WSS, and hybrid connection methods.

It is recognized that each attacker adheres to their unique "strategy" to successfully execute an attack. Even though attackers have their styles, some common actions they take can reveal their overall objective. Critical infrastructure systems are often targeted with well-known attack techniques. These include brute-force attempts to crack passwords, exploiting software flaws to remotely take control of devices, and launching malware attacks within the network once a foothold is gained.

To improve how honeypots lure attackers, we will be focusing on creating more intricate reward systems and crafting responses that are both believable and consistent. In the next phase, we will be expanding the honeypot to mimic an even wider range of IoT devices. We will also be deploying these improved honeypots across different public cloud platforms. Our primary objective is to assess the efficacy of employing machine learning methods in constructing a lambda function within a honeypot architecture to preemptively detect cyber-attacks prior to their widespread deployment across various cloud providers.

We are confident that our research on security solutions based on machine learning and deep learning represents a positive stride forward. It is poised to assist fellow academics and practitioners in discovering and deploying IoT security solutions in the future.

References:

Abdallah, W. (2023). Intrusion detection in IoT networks using machine learning techniques. *International Journal of Computers and Informatics (IJCI)*, *2*(8), 9-33. https://doi.org/10.59992/IJCI.2023.v2n8p1

Banaamah, A, & Ahmad, I. (2022). Intrusion detection in IoT using deep learning. *Sensors*, *22*(21), 8417. https://doi.org/10.3390/s22218417

Chen, D. et al. (2020). Cyber security in smart cities: A review of deep learning-based applications and case studies. *Sustainable Cities and Society*, *66*, 102655. https://doi.org/10.1016/j.scs.2020.102655

Fan, W. et al. (2018). Enabling an anatomic view to investigate honeypot systems: A survey. *IEEE Systems Journal*, *12*(4), 3906-3919. https://doi.org/10.1109/JSYST.2017.2762161

Géron, A. (2019). *Hands-on machine learning with scikit-learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.

Kasongo, S. M. (2021). An advanced intrusion detection system for IIoT based on GA and Tree Based Algorithms. *IEEE Access*, *9*, 113199-113212. https://doi.org/10.1109/ACCESS.2021.3104113

Liang, X., & Kim, Y. (2021). A survey on security attacks and solutions in the IoT Network. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 0853-0859. https://doi.org/10.1109/CCWC51732.2021.9376174

Merenda, M. et al. (2020). Edge machine learning for AI-enabled IoT devices: A review. *Sensors*, *20*(9), 2533. https://doi.org/10.3390/s20092533

Moustafa, N. et al. (2019). A holistic review of network anomaly detection systems: A comprehensive survey. *Journal of Network and Computer*, *128*, 33-55. https://doi.org/10.1016/j.jnca.2018.12.006

Salau, B. A. et al. (2022). Recent advances in artificial intelligence for wireless Internet of Things and cyber-physical systems: A comprehensive survey. *IEEE Internet of Things Journal*, *9*(15), 12916-12930. https://doi.org/10.1109/JIOT.2022.3170449

Sangra, P. et al. (2023). Energy efficiency in IoT-based smart healthcare. *Proceedings of Third International Conference on Computing, Communications, and Cyber-Security*, 503-515. https://doi.org/10.1007/9 78-981-19-1142-2_40

Sarwar, S. et al. (2022). Design of an advance intrusion detection system for IoT networks. *2nd International Conference on Artificial Intelligence (ICAI)*, 46-51. https://doi.org/10.1109/ICAI55435.2022.9773747

Sengupta, N. (2019). Designing security system for IoT. *2019 IEEE 5th International Conference on Big Data Security on Cloud (BigDataSecurity). IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, 195-199. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2019.00046

Sequeiros, J. B. F. et al. (2020). Attack and system modeling applied to IoT, cloud, and mobile ecosystems. *ACM Computing Surveys (CSUR)*, *53*, 1-32. https://doi.org/10.1145/3376123

Shafiq, M. et al. (2020). Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. *Future Generation Computer Systems*, *107*, 433-442. https://doi.org/10.1109/ACCESS.2019.2895334

Shone, N. et al. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(1), 41-50. https://doi.org/10.1109/TETCI.2017.2772792

Spitzner, L. (2020). The value of honeypots. Part one. Definitions and values of honeypots. *Symantec*. https://www.symantec.com/connect/articles/valuehoneypots-part-onedefinitions-and-values-honeypots.html

Tuna, O. F. et al. (2022). Exploiting epistemic uncertainty of the deep learning models to generate adversarial samples. *Multimed Tools Application*, *81*, 11479-11500, https://doi.org/10.1007.s11042-022-12132-7

Vinayakumar, R. et al. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access*, *7*, 41525-41550.

Vulnerability List: 2023. (2023). *Cve Details*. https://www.cvedetails.com/vulnerability-list/year-2023 /vulnerabilities.html/

Wang, M. (2017). *Understanding security flaws of IoT protocols through honeypot technologies: ThingPot-an IoT platform honeypot*. [Master's thesis, Delft University of Technology]. Netherlands.

Zhang, H. et al. (2021). Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. *Future Generation Computer Systems*, *122*. https://doi.org/10.1016/j.future.2021.03.024

Appendix

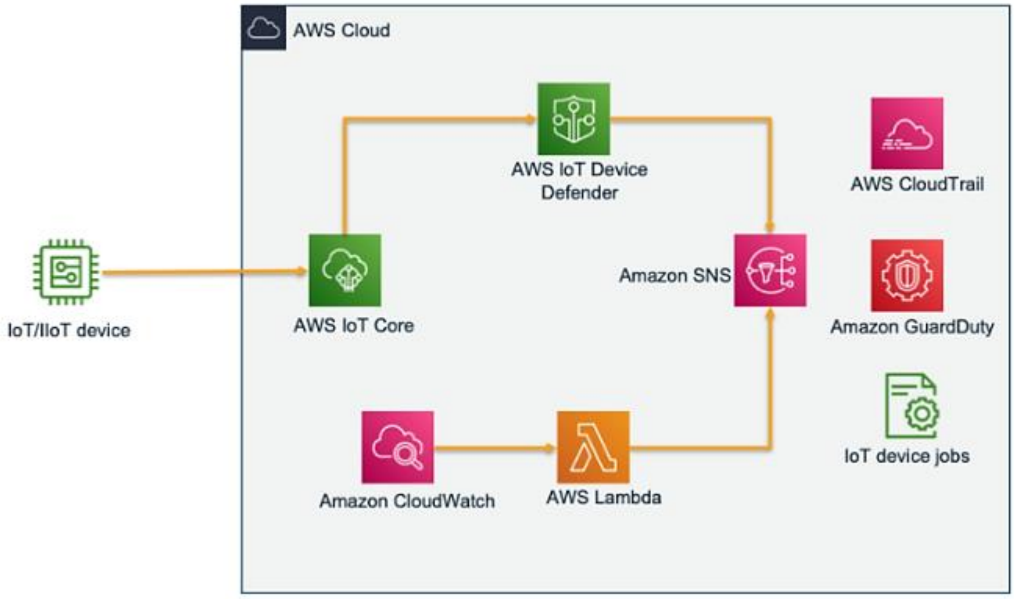

Figure 1. Solution architecture AISB

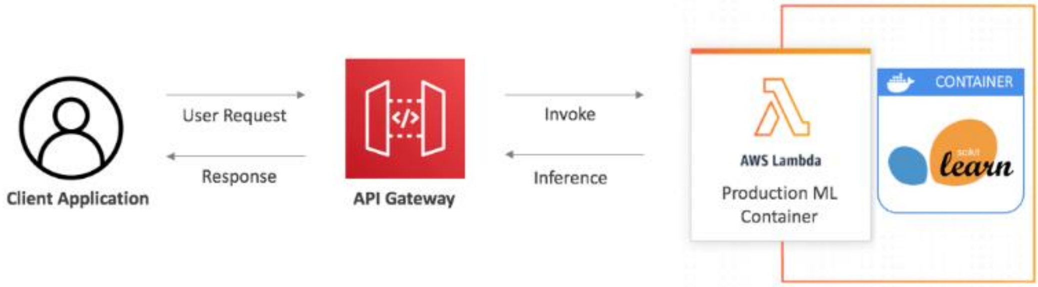

Figure 2. Architecture for ML inference

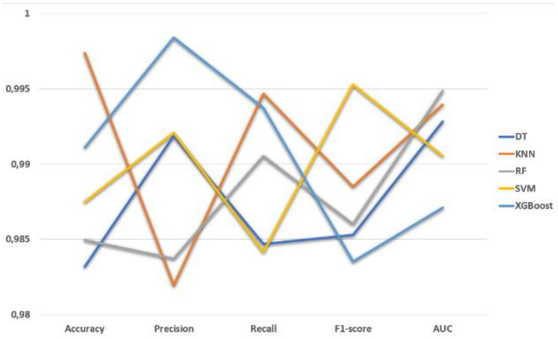

Figure 3. Comparison of different MLA efficiencies for detecting attacks on the MQTT

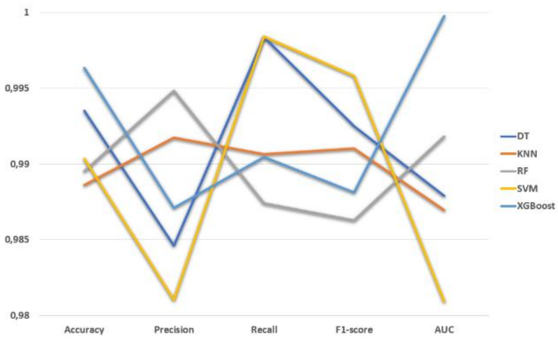

Figure 4. Comparison of different MLA efficiencies for detecting attacks on the HTTP

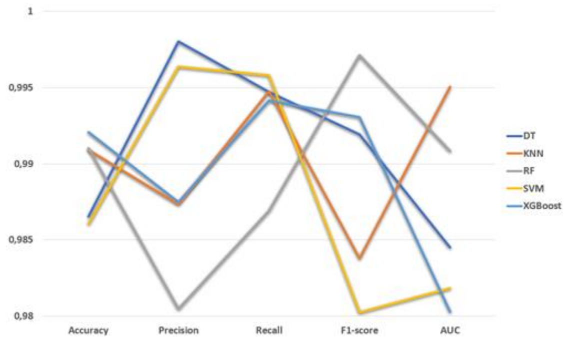

Figure 5. Comparison of different MLA efficiencies for detecting attacks on the MQTT over WSS

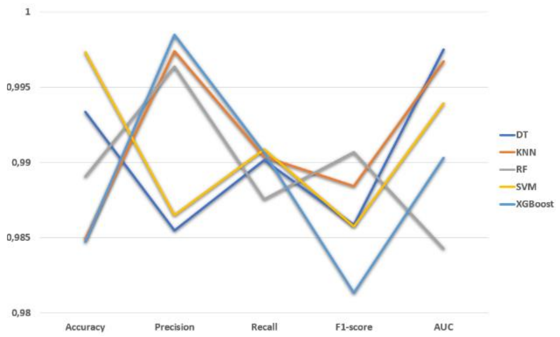

Figure 6. Comparison of different MLA efficiencies for detecting attacks on the hybrid connection method

Table 1. Security mechanisms of IoT communication protocols

Honepots	Open ports
MQTT	Simple User-name/password Authentication, TLS/SSL for data encryption
XMPP	SASL authentication, TLS/SSL for data encryption
AMQP	SASL authentication, TLS/SSL for data encryption
CoAP	DTLS/IPSEC
JMS	Vendor specific but typically based on TLS/SSL. Commonly used with JAAS API
SOAP	Address by WS-Security

Table 2. Evaluation metrics

Metric	Equation	Definition
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	Number of correctly classified samples divided by the total number of samples
Precision	$\frac{TP}{TP+FP}$	The fraction of detected attacks to total alarms
Recall	$\frac{TP}{TP+FN}$	Number of correctly classified attack samples divided by the total number of attack samples
F1-score	$2 \times \frac{Recall \ \times Precision}{Recall + Precision}$	The harmonic mean (weighted average) of the precision and recall
AUC	Area under the Receiver Operating Characteristics (ROC) curve	

Table 3. Modelling results of ML algorithms for detecting attacks on the MQTT

Model	DT	KNN	RF	SVM	XGBoost
Accuracy	0.98320	0.99739	0.98496	0.98750	0.99109
Precision	0.99187	0.98195	0.98371	0.99208	0.99840
Recall	0.98471	0.99468	0.99052	0.98417	0.99372
F1-score	0.98528	0.98851	0.98603	0.99529	0.98351
AUC	0.99284	0.99396	0.99485	0.99053	0.98709

Table 4. Modelling results of ML algorithms for detecting attacks on the HTTPS

Model	DT	KNN	RF	SVM	XGBoost
Accuracy	0.99351	0.98863	0.98953	0.99035	0.99634
Precision	0.98462	0.99174	0.99482	0.98106	0.98712
Recall	0.99837	0.99063	0.98740	0.99842	0.99045
F1-score	0.99249	0.99102	0.98627	0.99581	0.98813
AUC	0.98791	0.98694	0.99183	0.98093	0.99975

Table 5. Modelling results of ML algorithms for detecting attacks on the MQTT over WSS

Model	DT	KNN	RF	SVM	XGBoost
Accuracy	0.98654	0.99089	0.99104	0.98605	0.99209
Precision	0.99803	0.98732	0.98046	0.99638	0.98747
Recall	0.99470	0.99474	0.98693	0.99581	0.99414
F1-score	0.99193	0.98381	0.99711	0.98027	0.99306
AUC	0.98452	0.99504	0.99084	0.98184	0.98029

Table 6. Modelling results of ML algorithms for detecting attacks on the hybrid connection method

Model	DT	KNN	RF	SVM	XGBoost
Accuracy	0.99336	0.98493	0.98905	0.99732	0.98472
Precision	0.98549	0.99737	0.99638	0.98651	0.99849
Recall	0.99014	0.99039	0.98756	0.99088	0.99066
F1-score	0.98586	0.98843	0.99070	0.98574	0.98135
AUC	0.99750	0.99672	0.98428	0.99393	0.99030