Framework for detecting APTs based on steps analysis and correlation.

EKE, H.N., PETROVSKI, A., AHRIZ, H. and AL-KADRI, M.O.

2022

Framework for Detecting APTs Based on Steps Analysis and Correlation

H. N. Eke et al.

Abstract Advanced persistent threats (APTs) is an attack that uses multiple attack behaviour to penetrate a system, achieve specific targeted and highly valuable goals within a system. This type of attack has present an increasing concern for cyber security and business continuity. The resource availability, integrity and confidentiality of the operational cyber-physical systems' (CPS) state and control are highly impacted by the safety and security measures adopted. In this study, we propose a framework based on deep steps analysis and correlation of APTs approach, abbreviated as "APT-DASAC", for securing industrial control systems (ICSs). This approach takes into consideration the distributed and multi-level nature of ICS architecture and reflects on multi-step APT attack lifecycle. We validated the framework with three case studies: i) network transactions between a remote terminal unit (RTU) and a master control unit (MTU) within a supervisory control and data acquisition (SCADA) gas pipeline control system, ii) a case study of command and response injection attacks, and iii) a scenario based on network traffic containing hybrid of the real modern normal and the contemporary synthesized attack activities of the network traffic. Based on the achieved result, we show that the proposed approach achieves a significant attacks detection capability and demonstrates that attack detection techniques that performed very well in one application domain may not yield the same result in another. Hence, robustness and resilience of operational CPS state or any system and performance are determined by the security measures in place, which is specific to the application system and domain.

H.N. Eke
Robert Gordon University, Garthdee Road, Aberdeen, AB10 7GJ, e-mail: h.eke@rgu.ac.uk

A. Petrovski
Robert Gordon University, Garthdee Road, Aberdeen, AB10 7GJ e-mail: a.petrovski@rgu.ac.uk

H. Ahriz
Robert Gordon University, Garthdee Road, Aberdeen, AB10 7GJ e-mail: h.ahriz@rgu.ac.uk

M. O. Al-Kadri
Birmingham City University, Birmingham, B4 7XG e-mail: omar.alkadri@bcu.ac.uk

1 Introduction

Safety and security measures in place in terms of maintaining resource availability, integrity and confidentiality of the operational CPS state against cyber-threat such as APT remain one of the biggest challenges facing organizations and industries at various levels of operation [1].

The CPS systems are composed of computer and sub-systems that are interconnected based on the context within which an exchange of vital information through computer network takes place [2, 3, 4, 5]. CPS such as distributed control system (DCS) and SCADA contain control systems that are used in critical infrastructures such as nuclear power plants [6, 7], water, sewage and irrigation systems [8].

An APT, presented in Fig.1, is an attack that navigates around defences, breaches networks and evades detection, due to APTs stealthy characteristics and sophisticated levels of expertise and significant resources of contemporary attackers [9]. Whilst APTs have been attracting an increasing attention from the industrial security community, the current APTs best practices require a wide range of security countermeasures, resulting in a multi-layered defence approach that opens new research directions [12]. This type of attacks has drawn special attention to the possibilities of APT attacks on CPS, such as SCADA-based system. There have been few cases of successful attacks on ICS as recorded in [13, 14], these led to several attempts in developing methods to detect intrusions within network and isolated devices.

Most of these approaches focuses on detection of APT attack with respect to a specific domain. Work by author in [10] detects malicious PDFs based on whitelists and their compatibility as viable PDF files while study in [11] that focus on "Tokens" and utilises mathematical and computational analysis to filter spam emails focus on detection of only one step of APT lifecycle.

The computer systems used to control physical functions of the operating systems are not immune to the threat of today's sophisticated cyber-attacks and can be potentially vulnerable [15]. Potential threats can affect ICS devices at different level, hence security of each component within each level is extremely important to avoid compromise on any level [16].

APT attacks on a control system can be considered as stealthy disturbances, carefully designed with highly sophisticated combination of different techniques to achieve a specifically targeted and highly valuable goal by attackers [1]. These attackers are known to possess sophisticated levels of expertise and significant resources which allow them to create opportunities to achieve their objectives by using multiple attack vectors such as cyber, physical and deception. However, a well-designed control system may repel against external disturbances such as Reconnaissance. The unknown and dynamic nature of designed disturbance rules poses a security threat to CPS, which can be vulnerable to various types of cyber-attacks without any sign of system component failure [30]. Examples of these could be noticeable time-delays and serious control system degradation as a result of control systems been vulnerable to a denial-of-service (DoS) attack.

The successful removal or mitigating existing vulnerabilities, assessing whether a control system is experiencing any form of attack, and maintaining a secure and stable system state are the main CPS security.

1.1 Targeted APT attack on CPSs

APT attacks have affected many organizations as far back as 1998, with the first public recorded targeted attack named Moonlight Maze [31]. This Moonlight Maze attack targeted Pentagon, National Aeronautics and Space Administration (NASA), the US Energy Department, research laboratories and private universities by successfully compromised Pentagon computer networks, and accessed tens of thousands of file [32]. Past years has seen an increase in the number of organizations coming forward, admitting they have been targeted. Unfortunately, in the bid to protect organization's image and to avoid providing hackers with feedback, majority of those organization are not willing to share the attack details.

However, the four main recorded targeted attacks malware tailored against ICSs are STUXNET, BLACKENERGY 2, HAVEX and CRASHOVERRIDE [33, 34]. STUXNET is the first ever recorded attack aimed at disrupting physical industrial processes resulting in violation of system availability, while CRASHOVERRIDE is the second and also the first known to specifically target the electric grid [35, 14]. CRASHOVERRIDE is not unique to any vendor or configuration but utilises the knowledge of grid operations and network communications to cause disruptions resulting in electric outages [33, 36].

1.2 Safety of Cyber Physical Systems (CPSs)

CPS utilizes diverse communication platforms and protocols to increase efficiency and productivity. This is to reduce operational costs and further improve organization's support model [26]. The complexity of the ICS architecture and the increased efforts of controlling physical functions in processing and analysing data has leads to an intensified interactions between control and business networks [26, 27]. The possibility of deliberate targeted attacks as examined in [28] on control systems and the daily operational challenges due to this increased cyber physical interaction are on the high side [8, 27].

Ensuring the security of these systems are very important in order to avoid any operational disruption. However, this requires a complex approach to identify and mitigate security vulnerabilities or compromise at all levels within the ICS to maintain resource availability, safety, integrity and confidentiality, as well as becoming resilient against attacks [29]. We have suggested and implemented a multi-layered security model based on ensemble deep neural networks approach to secure ICSs.

The contribution of this paper can be summarised as follows:

- We discussed APT characteristics, lifecycle and give examples of the most significant confirmed cases of attack on CPS devices.
- We propose a novel approach using ensemble deep neural networks for realising multi-layered security detection for ICS devices. This approach takes RNNs variants to learn features from raw data in order to capture the malicious sequence patterns which reduce the cost of artificial feature engineering.
- We designed and implemented APT-DASAC - a multi-layered security detection approach - that takes into consideration the distributed and multi-level nature of ICS architecture and reflects on the four main SCADA-based cyber attacks. We further used stacked ensemble for APT-DASAC to combine networks' results for optimizing detection accuracy.
- A series of evaluation experiment, including individual APT step detection and attack type classification, were carried out. The achieved results suggest that the proposed approach has got the attack detection capability and demonstrated that performance of attack detection techniques applied can be influences by the nature of network transactions with respect to the domain of application.

1.3 Organization of Book Chapter

The remainder of this book chapter is organized as follows. Sect. 2 contains an overview of APT and APT lifecycle, brief discussion of related work directed toward the security of CPS. In Sect. 3, a detailed description of our proposed approach "architectural design of APT-DASAC" is discussed. The implementation of our APT-DASAC approach and the datasets used are discussed in Sect. 4. Experimental results are discussed in Sect. 5. Sect. 6 presents the conclusion of this book chapter.

2 Advanced Persistent Threats (APTs)

APTs and the actors behind them constitute a serious global threat. This type of attacks differs from common threats that seek to gain immediate advantage. APTs are broad in their targeting and processing. An APT is also very

- *resourceful*
- *with well defined objectives and purpose*
- *uses sophisticated methods and technology*
- *substantially* funded

2.1 Characteristics of APTs

An APT threat process follows a staged approach to target, penetrate and exploit its target. Understanding the advanced, sophisticated and persistent nature of APT is unavoidable in defending against such attacks.

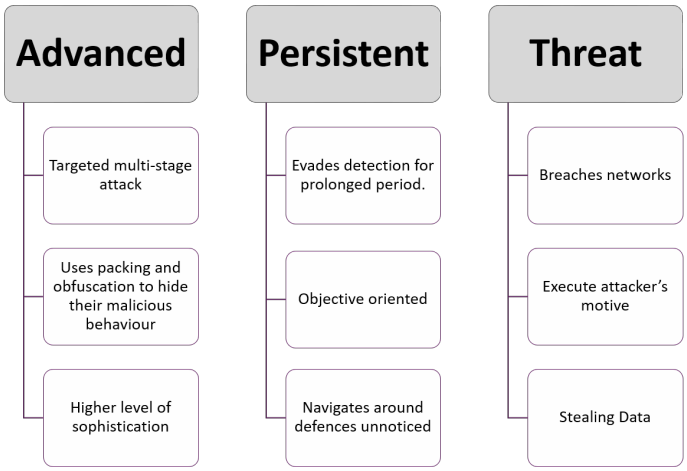


Fig. 1: Advanced Persistent Threats (APTs)

- ***Advanced -*** The advanced nature of APT provide the attackers with the capability of maintaining prolonged existence through stealthy approach inside an organization once they successfully breach security controls. Attackers uses sophisticated tools and techniques such as malware, if the malware is detected and removed, they change their tactics to secondary attack strategies as necessary [17].

- ***Persistent -*** The meaning of "Persistent" is expanded to persistently launching spear-phishing attacks against the targets by navigating a victim's network from system to system, obtaining confidential information, monitoring network activity, and adapting to be resilient against new security measures while maintaining a stealthy approach to reach its target [18].The mode of attack indicates the main functions of the APT-type malware, which usually placed more focus on spying instead of financial gain.

- ***Threat -*** The actors also have the capability of gaining access to electronically stored sensitive information. Other than the purpose of collecting national secrets or political espionage, based on the functions discovered, it is believed that this type of threats can also be applied to the cases in business or industrial espionage, spying acts or even un-ethical detective investigations [19, 20].

Examining the APT methods used to breach today's ICS security, it boil down to a basic understanding that attackers, especially those who have significant financial motivation, have devised an effective attack strategies centered on penetrating some of the most commonly deployed security controls. Most often it uses custom or dynamically generated malware for the initial breach and data-gathering step. The 'Advanced' and 'Persistent' are major features that differentiate APT from other cyber attacks.

2.2 Lifecycle of APTs Attack

APT attacks are generally known to utilise a zero-day exploits of unpublished vulnerabilities in computer programs or operating systems in combination with social engineering techniques. This is to maximise the effectiveness of the exploits that target unpatched vulnerabilities. Launching an APT attacks involves numerous hacking tools, a sophisticated pattern, high level knowledge, varieties of resources and processes. APTs proved extremely effective at infiltrating their targets and going undetected for extended periods of time, increasing their appeal to hackers who target businesses as highlighted in several large-scale security breaches [21, 22, 23].

Although each attack is customised with respect to attacker's target and aims at various stages of the kill chain, the patterns of APT attacks are similar in most cases but differ in the techniques used at each stage. For this study, we will describe six basic APT attack phases as used in our study, based on the literature review in combination with the "Intrusion Kill Chain (IKC)" model, described in [17, 24, 25].

1. ***Reconnaissance and Weaponization -*** This stage involves information gathering about the target. This could be, but not limited to, about organizational environment, employees' personal details, the type of network and defence target in use. The information gathering can be done through social engineering techniques, port scanning and open source intelligence (OSINT) tools.

2. ***Delivery -*** At this stage, attackers utilise the information gathered from reconnaissance stage to execute their exploits either directly or indirectly to the targets. In direct delivery, the attackers applies social engineering such as spear phishing by sending phishing email to target. While in indirect delivery, attacker will compromise a trusted third party, which could be a vendor or frequently visited website by the target and uses these to deliver an exploit.

3. ***Initial Intrusion and Exploitation -*** At this stage, attacker gain access to target's network by utilising the credential information gathered through social engineering. The malware code delivered at this stage is downloaded, installed and activate backdoor malware, creating a command and control (C&C) connection between the target machine and a remote attacker's machine. Once a connection to the target machine has been secured, the attacker continues to gather more relevant

information such as security configuration, user names, sniff passwords from target network while maintaining a stealthy behaviour in preparation for next attack.

4. ***Lateral Movement and Operation -*** At this stage, once the attacker establishes communication between the target's compromised systems and servers, the attacker moves horizontally within the target network, identify the servers storing the sensitive information on users with high access privileges. This is to elevate their privileges to access sensitive data. This make their activities undetectable or even untraceable due to the level of access they have. Attackers also create strategy to collect and export the obtained information.

5. ***Data Collection -*** This stage involves utilising the privileged users credentials captured during the previous stage to gain access to the targeted sensitive data. With the attackers having a privileged access, they will now create redundant copies of C&C channels should there be any change in security configuration. Once the target information has been accessed, redundant copies are created at several staging points where the gathered information is packaged and encrypted before exfiltration.

6. ***Exfiltration -*** At this stage, once an attacker has gained full control of target systems, they proceed with the theft of intellectual property or other confidential data. The stolen information is transferred to attackers' external servers in the form of encrypted package, password protected zip files, or through clear web mail. The idea of transferring information to multiple servers is an obfuscation strategy to stop any investigation from discovering the final destination of the stolen data.

2.3 Related Work

Diverse approaches have been proposed and successfully implemented to address different types of attacks. These proposed methods have led to a significant pool of solutions geared towards addressing security and resilience of CPS devices. Most of these approaches focuses on detection of attack with respect to a specific domain.

2.3.1 Attack Detection

One of this detection model is intrusion cyber kill chain (IKC). This was created by Lockheed Martin analysts in 2011 to support a better detection and response to attacker's intrusions by applying the IKC model to describe different stages of intrusion [25, 37]. Although, this model is not directly applicable to the ICS-custom cyber attacks, it serves as a great building foundation and concept to start with [25]. Few other approaches in the literature includes, but not limited to, the attack

detection based on communication channels, a notion of stealthiness, false data injection attacks (FDI) and network information flow analysis.

Work in [38], made use of the possibility of unprotected communication channels for sensor and actuator signals in plant, which may allow attackers to potentially inject false signals into the system. The authors model an approach to capture the vulnerabilities and the consequences of an attack on the ICSs, being focused on "The closed-loop control system architecture", where the plant is controlled by the supervisor through sensors and actuators in a traditional feedback loop. Their approach aims at detecting an active online attack and disables all controllable events after detecting the attack, preventing thereby the system from reaching a pre-defined set of unsafe states. This work is a complementary study to another work in [39], where the authors investigated an online active approach using a multiple-supervisor architecture that actively counteracts the effect of faults and introduces the idea of safe controllability in active fault tolerant systems to characterize the conditions that must be satisfied when dealing with the issue of fault tolerance.

Other proposed approaches that mainly focus on APT detection based on network information flow analysis that is not specific for CPS as reviewed for this work include an APT attack detection method based on deep learning using information flows to analyzed network traffic into IP-based network flows, reconstruct the IP information flow and uses deep learning models to extract features for detecting APT attack IPs from other IPs [46]. The authors in [47], propose an approach to detect the hidden C&C channel of unknown APT attacks using network flow-based C&C detection method as inspired from the belief that: (i) different APT attacks share the same intrusion techniques and services, (ii) unknown malware evolves from existing malware, and (iii) different malware groups share the same attributes resulting to hidden shared features in the network flows between the malware and the C&C server within different attacks. They applied deep learning techniques to deal with unknown malicious network flows and achieved an $f1-score$ of 96.80%.

2.3.2 Attack Mitigation

Authors in [40] considered a notion of stealthiness for stochastic CPS that is independent of the attack detection algorithm to quantify the difficulty of detecting an attack from the measurements. With the belief that the attacker knows the system parameters and noise statistics, and can hijack and replace the nominal control input by characterizing the largest degradation of Kalman filtering induced by stealthy attacks. The study reveals that the nominal control input is the only critical piece of information to induce the largest performance degradation for right-inverting systems, while providing an achievable result that lower bounds of performance degradation that an optimal stealthy attack can achieve within non-right-inverting systems. While Miloševič et al in [41] examined the presence of bias injection attacks for state estimation problem for stochastic linear dynamical system against the Kalman filter as an estimator equipped with the chi-squared been used as a detector

of anomalies. This work suggests that that the issue of finding a worst-case bias injection attack can be controlled to a certain degree.

Also, Xu et al [42] focus on a stealthy estimation attack that can modify the state estimation result of the CPS to evade detection. In their study, the chi-square statistic was used as a detector. A signaling game with evidence (SGE) was used to find the optimal attack and defense strategies that can mitigate the impact of the attack on the physical estimation, guarantying thereby CPS stability.

Furthermore, study on industrial fault diagnosis using deep Boltzmann machine and multi-grained scanning forest ensemble was done by [43] and FDI [6]. Also, the possibility of accurately reconstructing adversarial attacks using estimation and control of linear systems when sensors or actuators are corrupted [44], is studied in the quest for CPS security and more resilience against targeted attacks. The authors in [45] considered the case of the FDI attacks detection issue as a binary classification case and propose a statistical FDI attacks detection approach based on a new dimensionality-reduction method using a Gaussian mixture model and a semi-supervised learning algorithm to examine the coordinates of the data under the newly orthogonal axes obtained to establish FDI attacks if the outputs of the Gaussian mixture model exceed the pre-determined threshold.

3 APT Detection Framework

In this section we present the description of our proposed APT-DASAC framework architectural design for APT intrusion detection. APT attacks purposefully launched to target critical infrastructures, such as SCADA network as highlighted in [9], is a multi-step attack. The detection of a single step of an APT itself does not imply detecting an APT attack [1]. Hence, APT detection systems should be able to detect every single possible step applied by an APT attacker during the attack process.

3.1 Architectural Design of APT-DASAC

The design of our proposed model for APT intrusion detection system (IDS) is built to run through three stages. This involves implementing a multi-layered security detection approach based on Deep Leaning (DL), that takes into consideration the distributed and multi-level nature of the ICS architecture and reflect on the APT lifecycle for the four main SCADA cyber-attacks as suggested in [6].

The implementation of our design model shown in Fig. 2 consist of three stages.

Stage 1: Data input and probing layer
Stage 2: Data analysis Layer
Stage 2: Decision Layer

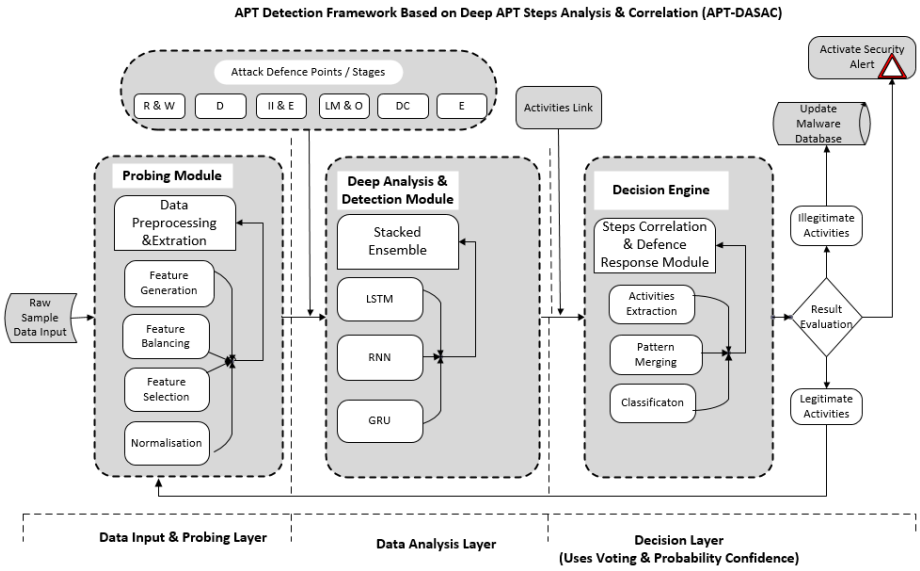


Fig. 2: Detection framework based on deep APT step analysis and correlation (APT-DASAC)

3.2 Three Layers of APT-DASAC

The processes taken to implement our proposed model "APT-DASAC" is discussed as follows: -

For the purpose of this model explanation and illustration, the New Gas Pipeline (NGP) and University of New South Wales (UNSW-NB15) datasets were used. The specific step-by-step pseudocode for APT-DASAC and the detection process are described in the following subsection.

The first stage of this approach, *"Data input and probing layer"* involves data gathering and pre-processing sample data by transforming the data into an appropriate data format ready to be used in the second stage *"Data analysis Layer"*. This second stage applies the core process of APT-DASAC, which takes stacked recurrent neural network (RNN) variant to learn the behaviour of APT steps from the sequence data. These steps reflect the pattern of APT attack steps. In the final stage *"Decision Layer"*, we use ensemble RNN variants to integrate the output and make a final prediction result.

3.2.1 Step-by-Step Pseudocode for APT-DASAC Layers:

The experimental implementation pseudocode of our proposed framework in Fig. 2 is represented by Algorithm 1-3 as used to build the proposed model.

- *Pseudocode for data pre-processing*
- *Pseudocode for data analysis*
- *Pseudocode for detection and prediction process*

The ***pre-processing data*** stage takes raw network traffic data as an input from a specific problem domain, processes and transforms the data into a meaningful data format that the algorithm requires by converting any symbolic attributes into usable features, and deals with null values using ***Step 1 to Step 7c in Algorithm 1***. The output from this stage is a ***new transformed data*** containing valuable information that the analyses stage will utilize.

3.2.2 Data Input and Probing Layer

This layer consists of two modules; (i) Data Input and (ii) Probing Module. Algorithm 1 shows the steps for this module process.

1. ***Data Input*** - involves data gathering, raw sample/simulated synthetic data been introduced into the system and transfer the collected data to probing module.
2. ***Probing Module*** – involves data pre-processing and feature transformation which runs through four stages. Here all the data that has been collected and introduced into the module are encoded into numerical vector by the pre-processor ready to go through the neural network.
   a. ***Feature Transformation***: UNSW-NB15 dataset consists of 42 features with three of these features been categorical (proto, service and state) data. These three features need to be encoded into numeric feature vector as it goes to the neural network for analysis, classification, detection and prediction. For this reason, Pandas $get_dummies()function$ was used, this function creates new dummy columns for each individual categorical feature. This leads to increase in the number of columns from 42 to 196 features available for onward analysis.
   b. ***Balancing Training & Testing Data Features***: Both training and testing data contains different number of categorical features, this implies that $get_dummies()function$ will generate different number of columns for training and testing data. However, the number of features in both sets need to be the same. In this case, we deployed $set().union()function$ to balance the training and testing datasets.
   c. ***Normalization***: At this stage, the $ZScore$ method of standardisation is used to normalize all numerical features to preserve the data range, to introduce the dispersion of the series, and to improve model convergence speed during training.

Algorithm 1: Data Input and Probing Layer Pseudocode

```
1 - Pseudocode for Data Pre-processing
    Step 1: Input the sample dataset.
    Step 2: Convert the symbolic attributes features.
    Step 3: Return new set of data
    Step 4: Separate the instances of dataset into classes
            (y)
    Step 5: Scale & normalize data (x_(t)) into values from
            [0 to 1]
    Step 6: Split dataset into training and testing data
        Step 7a: Balance & reshape the Training & Testing
                 data features
        Step 7b: Return balanced & reshaped Training &
                 Testing data
        Step 7c: Pickle transformed data into a byte stream
                 and store it in a file/database (.pki)
```

3.2.3 Analysis Layer

The rate of attack detection is affected by the parameters used as these parameters have direct impact on attack detection. Based on this, several experiments with different network configuration were implemented to find the best optimal values for parameters such as learning rate and network structure.

Also, to achieve a good detection rate for rare attack steps whilst maintaining overall good model performance, two issues need to be considered - the rare attack class distribution and the difficulty of correctly classifying the rare class. When considering the class distribution, more emphasis should be placed on the classes with fewer examples. Secondly, more emphasis should be given to examples that are difficult to be correctly classified.

At this layer, the processed data are used to build a model that analyses and distinguishes attack(s) from normal activities, taken note of the identified issues with class distribution and classification of rare attacks. The result of this layer is passed to Decision Engine layer.

3.2.4 Decision Layer

This Layer operates using three approaches: firstly, it receives information from the analysis layer and extract the attack step present. Secondly, it processes this information and links it to the related attack steps. Lastly, it uses voting and probability confidence to establish if the attack is a potential chain of attack campaign is found, and if it is consistent with other attack campaigns.

Algorithm 2: Analysis Layer Pseudocode

```
1 - Pseudocode for Sequence Data Training and Testing
    During the training and testing stage, steps 8a - 8e
    are followed in each iteration.

    Step 8: Train the model with this new training dataset.
        Step8a: Sequentially fetch a sample data (x_(t))
                from the training set.
        Step8b: Estimate the probability (p) that the
        example should be used for training.
        Step8c: Generate a uniform random real number μ
                between 0 and 1.
        Step8d: If μ <p, then use x_(t)to update the RNN by
                Equation (5) for any training sample (x_(i) y_(i)).
        Step8e: Repeat steps 1-4 Algorithm 1 until there is no
                sample left in the training set.
    Step 9: Test model with testing data from Step 7b
    Step10: Compute and evaluate the Model performance
            accuracy output - classification, detection
            and prediction
```

Algorithm 3: Decision Layer Pseudocode

```
1 - Pseudocode for Analysis, Detection and Prediction
    In analysis detection and prediction stage, steps
    11 - 17c are followed in each iteration.
    Step11: Set ip_units, lstm_units, op_units and
            optimizer to define LST Network (DL)
    Step12: Fetch the processed data (x_(i))
            #pre-processed data through Steps 1-7(Algorithm 1)
    Step13: Select specified training window size (tws)
            and arrange x_(i) accordingly.
    Step14a: for n_epochs and batch_size do #each iteration
           Step14b: Takes the input vector within specified
                    training window size (x_(tws)) at time
                    (t) together with previous information,
                    initially set to 0
       Step14c: Train the Network (L) x_(tws+1))
    Step14d: end for
    Step15: Run Predictions using L
    Step16: Calculate the categorical_loss_function L(o,y)
            using Equation (11).
    Step17: Output result
       Step17a: percentage detection rate of individual
                attacks detected
       Step17b: Overall detection rate
       Step17c: Confirmation if there is any existence
              or complete APT steps (full APT scenario)
```

3.2.5 Attack Step Impacts

The attack impact is determined at this stage through the decision engine by correlating the output from the analysis layer using probability confidence to check for any presence of security risks. If an attack or security risk is present, it requests the defence response module to raise a security alert. This is checked with the previously detected step to see if this could be related to the newly discovered security risk alert. This is to reconstruct APT attack campaign steps, and hence highlights an APT campaign scenario so that an appropriate action can be taken.

The impact of an attack can be considered as low depending on the attack activity stage. However, if this stage can be linked with other attacks step to show that it is part of that attack campaign, forming a full APT step cycle, then the impact at this stage can be considered as high. With this information in mind an appropriate response can be taken.

4 Implementation of APT-DASAC Approach

In this section we describe the platform and the approach taken to implement the APT-DASAC. These includes the implementation setup, the hyperparameters settings used, and the datasets used.

4.1 Implementation Setup

The ensemble RNN-based attack detection models as explained in [6] were implemented. The network topology and payload information values of the NGP dataset containing 214,580 Modbus network packets with 60,048 packets that are associated with cyber attacks were used. These attacks are placed into 7 different categories with 35 different specific attack types as explained in [48, 49]. These attacks categories align with APT lifecycle. Fig. 3 and 4 shows the number of records in each of the categories and the main four types of attacks as contained in the NGP data. During the experimental setup, the first task was focused on deriving hyper-parameter values for best performance model. Secondly, the best hyperparameter values were implemented in measuring the model performance.

The standard data mining processes such as data cleaning and pre-processing, normalization, visualisation and classification were implemented in Python. The batch size of 124 to 300 epochs are run with a learning rate set in the range of 0.01-0.5 on a GPU-enabled TensorFlow network architecture. All the 17 features were used as input vector with 70% as training set and 30% as validation set for the multi-attack classification. The training dataset were normalized from 0 to 1. This was trained using sigmoid activation function through time with ADAM optimizer, sigmoid function was used on all the three gates and categorical cross-entropy as loss

function for error rate. Also, these tasks were carried out with traditional machine learning (ML) classification algorithms - Decision Tree (DT). The ML classification result was compared to stacked Deep ensemble RNNs-LSTM result in order to further evaluate the APT steps detection capability of the experimental approach. Result evaluation is discussed in Sect. 5.

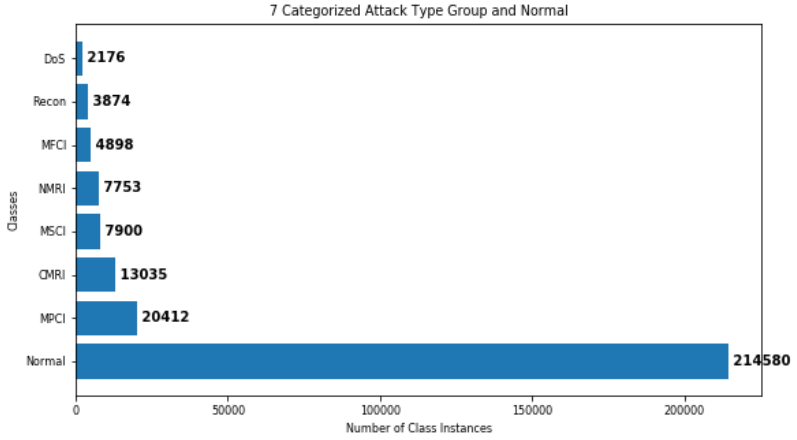


Fig. 3: NGP dataset records

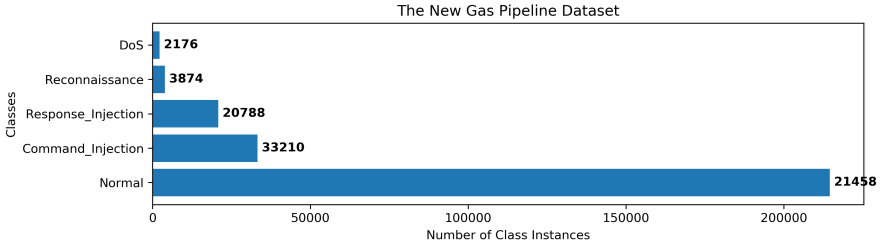


Fig. 4: Four main attack group and normal classes

4.1.1 Hyperparameters Settings

- Batch sizes: 64 and 128
- Learning rate: 0.0002 to 0.00005 with polynomial decay over all the epochs.
- Epochs: 100 – 300 epochs.
- Neural network: Four layers were used
- Each of the hidden layers has a ***sigmoid/ReLU*** activation function applied to it to produce non-linearity. This transforms the input into values usable by the output layer.
- The ***softmax*** function is applied to the output layer to get probabilities of categories. This also helps in learning with ***cross-entropy loss*** function.
- Adaptive Moment Estimation ***(Adam)*** optimizer is used for the back propagation to minimise the loss of categorical-cross entropy.
- The ***dropout*** is used to alleviate the over-fitting (used as regularization technique used to prevent over-fitting in Neural Networks. This randomly removes the units along with connections.

4.2 Implementation Dataset

Due to the specific dynamic nature of APT attack, that does not follow a unique pattern, availability and accessibility of dataset containing realistic APT scenario have became a challenging issue when testing and comparing APT detection models. For the implementation of our approach, the NGP[1] and UNSW-NB15[2] datasets were used. Both datasets are available for research purposes.

4.2.1 New Gas Pipeline Dataset (NGP) Explained

The NGP data is generated through network transactions between a RTU and a MTU within a SCADA-based gas pipeline at Mississippi State University. This data was collected by simulating real attacks and operator activity on a gas pipeline using a novel framework for attack simulation as described in [48] and [52]. The data contains three separate main categories of features – the network information, payload information, and labels.

The *network topologies* and the *payload information* values of SCADA systems are very important to understand the SCADA system performance and detecting if the system is in an out-of-bounds or critical state [3]

[1] https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets

[2] https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

[3] http://www.simplymodbus.ca/TCP.htm. Accessed on 10/03/2021

4.2.2 Three Main Features of NGP dataset

- ***Network Information*** - This category provides a communication pattern for an IDS to train against. In SCADA systems, network topologies are fixed with repetitive and regular transactions between the nodes. This static behaviour favours IDS in anomalous activities detection.
- ***Payload Information*** - This provides an important information about the gas pipeline's state, settings, and parameters, which helps to understand the system performance and detecting if the system is in a critical or out-of-bounds state.
- ***Labels*** – is attached to each line in data to indicate if the transaction within the system activity is normal or malicious activities.

4.2.3 Identified Cyber Threats in NGP dataset

The original gas pipeline data as in [49] was improved to create a new NGP data by;

- *parameterizing* and *randomizing* the order in which the attacks were executed
- executing *all the attacks* as contained in the original data created by Gao [49]
- implementing all the attacks in a *man-in-the-middle* fashion
- to includes all the *four types of attacks* as shown below

  - ***Interception*** - In this type of attack, attacks are sent to both the attacker and to the initial receiver. These types of attacks enable gaining system information such as normal system operation, each protocols node, the brand and model of the RTUs that the system is using.
  - ***Interruption*** - this type of attack is used to block all communication between two nodes in a system - e.g., DoS between the MTU and an RTU slave device in the gas pipeline.
  - ***Modification*** - This type of attacks allows an attacker to modify parameters (set point parameter exclusively and leave all other parameters untouched) or states in a system, such as the gas pipeline.
  - ***Fabrication*** - attackers execute this type of attack creating a new packet to be sent between the MTU and RTU.

4.2.4 Raw Dataset

In this sub section, we will use Fig. 5 to describe and illustrate the instances futures as contained within the NGP dataset.

- ***The first feature*** - represent the Modbus frame as received either by the master or slave device. All valuable information from the network, state, and parameters of the gas pipeline are also contained in this Modbus frame.
- ***The second and third feature*** - represent the attack category and specific attack that was executed. In case of Modbus frame normal operation, both of these

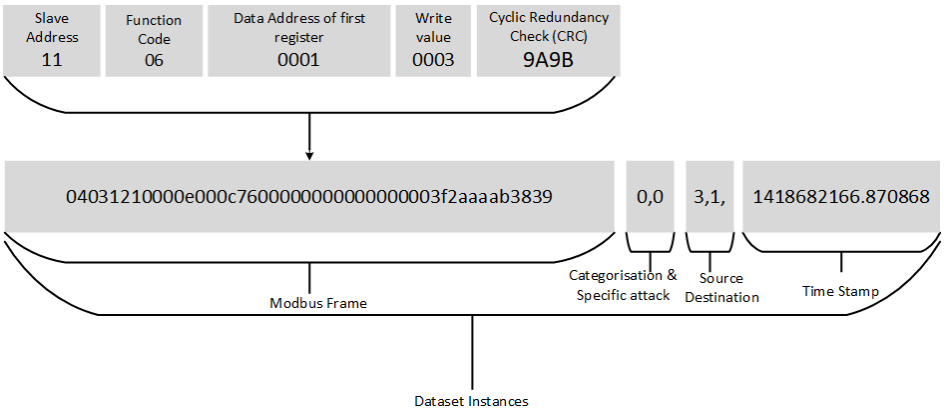


Fig. 5: The instances within NGP raw dataset

features will report a zero. Both are useful to train a supervised learning algorithm, as they allow the algorithm to learn the behaviour of these attack patterns.

- ***The fourth and fifth features*** - represent the source and destination of the frame. There are only three possible values for the source and destination feature. The value can be a *'1'*, indicates that the master device sent the packet, *'2'*, meaning the man-in-the-middle computer sent the packet, or *'3'*, indicates that the slave device sent the packet.
- ***The last feature (6th)*** - in the raw data contains a time stamp which can be used to calculate the time interval between change. In system normal operation, slight change may be observed between time intervals, however any modification or malicious activity such as malicious command injection may lead to noticeable time interval change.

4.2.5 Cyber-attacks as contained in the NGP Dataset Record

The NGP data contains 214,580 Modbus network packets with 60,048 packets associated with cyber attacks. Each record contains 17 features in each network packet. These attacks are placed into 7 different attack categories with 35 different specific type of attacks. These attacks categories and the individual specific attack as represented in Fig. 3 and Table 1 will be used to demonstrate an APTs steps detection with our proposed APTs detection framework in line with APTs life cycle as described in [9].

These 7 attacks categories are further grouped into 4 overall categories to align with APT life cycle and the 4 identified types of cyber attacks as described below.

- ***Response injection attacks*** – contains two types of attacks, naïve malicious response injection (NMRI) (which occur when the malicious attacker do not have sufficient information about the physical system process) and complex malicious

response injection (CMRI) (these type of attack designs attacks that mimic certain normal behaviours using physical process information making it more difficult to detect).

- ***Command injection attacks*** - contains three attacks, malicious state command injection (MSCI), malicious parameter command injection (MPCI) and malicious function code injection attacks (MFCI). These attacks inject control configuration commands to modify the system state and behaviour, resulting to: (a) loss of process control, (b) device communication interruption, unauthorized modification of (c) process set points and (d) device control.
- ***DoS attacks*** – disrupt communications between the control and the process through interruption of wireless networks or network protocol exploits.
- ***Reconnaissance*** – collects network and system information through passive gathering or by forcing information from a device.

Table 1: Attack categories with normal records type

Attack Categories	Abbreviation	Values	APTs Step
Normal	Normal	0	Not Applicable
Naïve Malicious Response Injection	NMRI	1	Delivery
Complex Malicious Response Injection	CMRI	2	Exploitation, Exfiltration
Malicious State Command Injection	MSCI	3	Data Collection, Exploitation
Malicious Parameter Command Injection	MPCI	4	Data Collection, Exploitation
Malicious Function Code Injection	MFCI	5	Data Collection, Exploitation, Exfiltration
Denial of Service	Dos	6	Data Collection, Exploitation, Exfiltration
Reconnaissance	Recon	7	Reconnaissance

4.2.6 UNSW-NB15 Dataset

UNSW-NB15 dataset as representation in Fig. 6 and 7 was created by Australian Centre for Cyber Security (ACCS)[4] in their Cyber Security Lab. A hybrid of the modern normal and abnormal network traffic features of UNSW-NB15 data was created using the IXIA PerfectStorm tools[5] to simulate nine families of attacks categories as follows: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. In other to identify an attack on a network system, a comprehensive dataset that contains normal and abnormal behaviours are required to carry out a proper evaluation of network IDS effectiveness and performance [55]. Hence, the UNSW-NB15 dataset [56] was chosen for this study as the IXIA PerfectStorm tool used to generate the data contains all information about new attacks

[4] https://www.unsw.adfa.edu.au/unsw-canberra-cyber

[5] https://www.ixiacom.com/products/perfectstorm

on CVE website[6], which is the dictionary of publicly known information security vulnerability and exposure and are updated continuously as stated in [56].

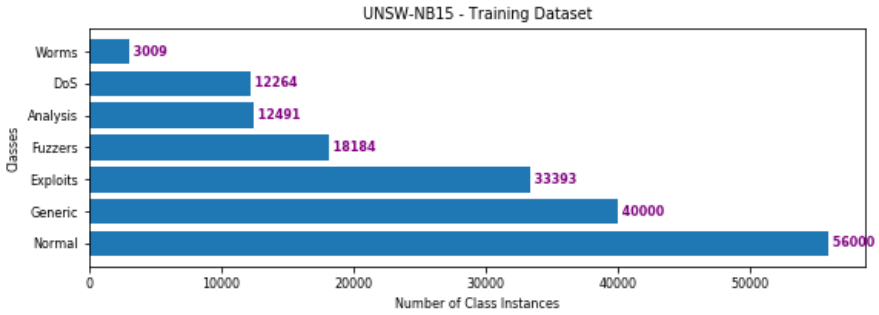


Fig. 6: UNSW-NB15 train dataset

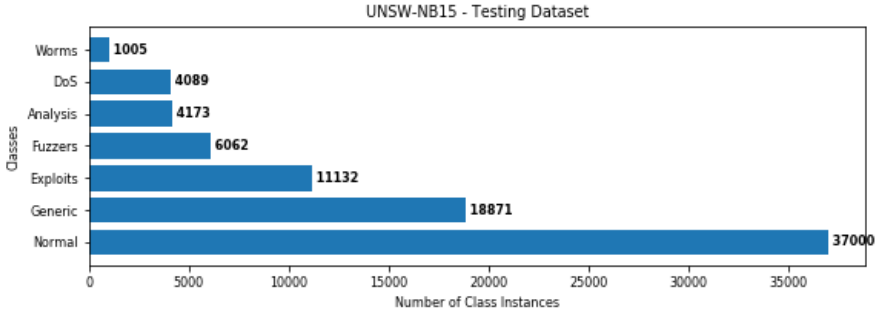


Fig. 7: UNSW-NB15 test dataset

5 Experimental Evaluation of APT-DASAC Approach

Generally, accuracy is used as a traditional way of measuring classification performance. This metric measure is no longer appropriate when dealing with multi-class imbalance data since the minority class has little or no contribution when compared to majority classes toward accuracy [57]. For these reasons, we applied synthetic minority oversampling technique (SMOTE) for handling data imbalance as explained in [1].

[6] https://cve.mitre.org/

Evaluation Metrics: We used *precision, recall, f1-score, overall accuracy*, area under the curve (AUC) receiver operating characteristic (ROC) and *confusion matrix* to validate the performance of implementing APT-DASAC for attack detection and clearer understanding of the output.

5.1 Result and Discussion

In our previous study [6], we implemented a DL multi-layered security detection approach which focused on detecting command injection (CI) and response injection (RI) attacks. We noticed a higher detection rate of CI to RI, although CI has more connection records and obtained a significant detection rate with 0% False Positive Rate (FPR) and True Positive Rate (TPR) of 96.50%. Based on the outcome of our analysis, We arrived on the conclusion that performance of attack detection techniques applied can be influences by the nature of the network transactions with respect to the domain of application and made suggestion for further investigation in different domain.

We acknowledge the need to investigate this further in other to ascertain this claim. We implemented the application of stacked ensemble-LSTM variants for APT-DASAC. This approach combines networks' results as to optimize attack detection rate. To validate this approach for detecting APT step attacks, statistical metrics such as *precision, recall, f1-score, AUC-ROC and overall accuracy* are calculated (i) to evaluate the ability of this approach to accurately detect and classify an abnormal network as an attack, (ii) to check the ability of this model to detect different type of attacks accurately, and (iii) to get a clearer understanding of the output.

Fig. 8 and 9 contains the statistical classification report obtained from implementing deep ensemble-LSTM variants and ML-DT on NGP dataset respectively. These reports shows that our approach achieved an average P, R and $f1$ of 88%, 86% and 82% respectively with overall detection accuracy of 85% and macro f1 of 62%, while the implemented ML-DT obtain 95% for P, R and $f1$ with overall detection accuracy of 94% in detecting attacks.

Considering the fact that the proposed approach detects APT step activities in different stages, we generated ROC curves score for the stages as shown in Fig.10. The average of the 5 steps curves is evaluated and consolidated into a single graph representing their respective $AUCcurve$ and obtain micro-average ROC curve area of 91% and macro-average ROC curve area of 72%. It is evident from Fig.10 that the classification of APT attack detection in class 3 stage has the ROC curve area of 93% , this is largely attributed to the number of connection record exhibited in this stage, while the class 4 stage has the lowest ROC curve area of 51%. Our proposed approach seems to achieve a good performance since the weighted average of the ROC curve area is closer to 1. A high area under the curve represents both high recall

and high precision, an ideal model with high precision and high recall will return many results, with all results labelled correctly.

The results shown in Fig. 11 and 12 are the visual representation of each algorithm's validation accuracy and loss rate on each epochs. There are some spikes in the validation accuracy and loss, following the individual model accuracy and loss per epoch, achieving training and validation accuracy of 85.59%, 85.88% with validation loss of 33% for LSTM; 85.97%, 85.16% with validation loss of 35% for RNN and 86.13%, 85.71% with validation loss of 34% for GRU. It what to note that the value of training and validation accuracy are quite close to each other, indicating that the model is not overfitting with overall average mean detection accuracy and validation average accuracy of 85%.

We also implemented the same approach with UNSW-NB15 data, the average detection accuracy of 93.67% as recorded in Table 2, which is slightly higher than 85% obtained when NGP data was implemented.

	precision	recall	f1-score	support
Command_Injection	0.97	0.51	0.67	10959
DoS	0.99	0.44	0.61	718
Normal	0.85	1.00	0.92	70812
Reconnaissance	0.94	0.93	0.94	1279
Response_Injection	1.00	0.02	0.03	6860
avg / total	0.88	0.86	0.82	90628

Fig. 8: Classification - report for ensemble-LSTM variants on NGP dataset

	precision	recall	f1-score	support
Command_Injection	0.98	0.96	0.97	10959
DoS	0.96	0.93	0.95	718
Normal	0.96	0.97	0.97	70812
Reconnaissance	0.98	0.97	0.98	1279
Response_Injection	0.72	0.67	0.69	6860
avg / total	0.95	0.95	0.95	90628

Fig. 9: Classification - report for ML-DT on NGP dataset

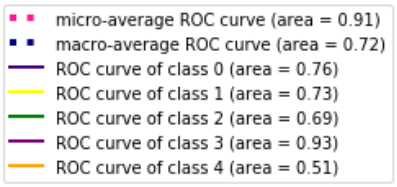
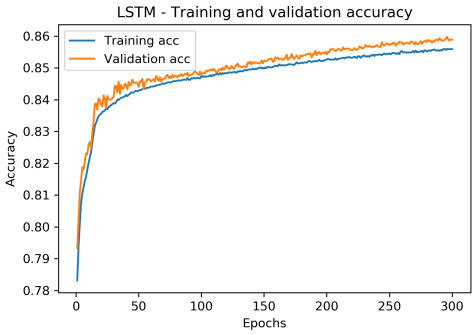


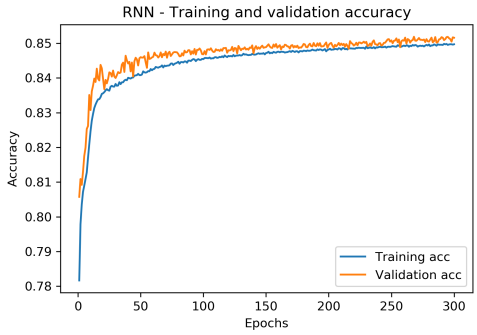
Fig. 10: AUC-ROC - report for ensemble-LSTM variants on NGP dataset

Table 2: Performance report for ensemble-LSTM variants on UNSW-NB15 dataset

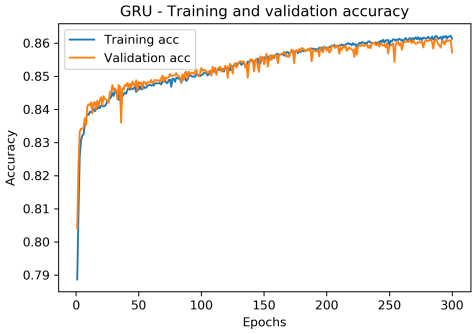
Algorithm	Average Accuracy	Validation Accuracy	Validation loss
LSTM	93.74%	82.29%	21.82%
RNN	92.88%	81.43%	20.50%
GRU	94.41%	82.11%	20.46%
ensemble-LSTM variants	93.67%	84.94%	20.47%



(a) Accuracy validation against epochs for LSTM

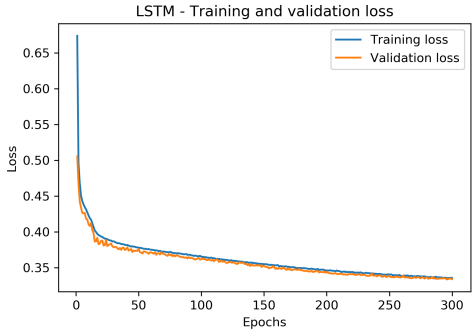


(b) Accuracy validation against epochs for RNN

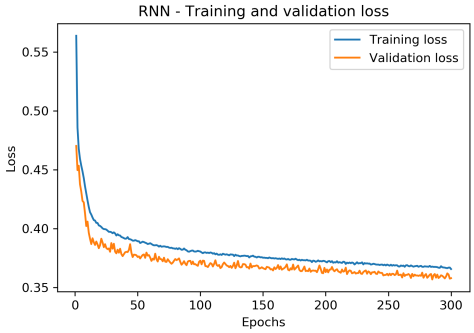


(c) Accuracy validation against epochs for GRU

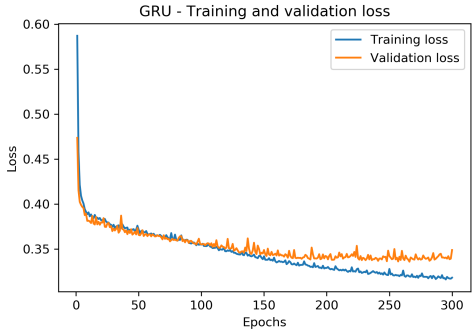
Fig. 11: Validation accuracy against epochs on NGP dataset



(a) Validation loss against epochs for LSTM



(b) Validation loss against epochs for RNN



(c) Validation loss against epochs for GRU

Fig. 12: Validation loss against epochs on NGP dataset

5.1.1 Our Proposed Approach and other Works on APTs Detection

Few proposed APT detection approach recorded in Table 3 as reviewed for this paper includes, work in [46], an APT attack detection method based on Bidirectional Long Short-Term Memory (BiLSTM) and Graph Convolutional Networks (GCN) to analyzed network traffic into IP-based network flows. This approach achieved 98.24% of normal IPs and 68.89% of APT attack IPs using Malware Capture CTU-13 data warehouse dataset. The authors in [47], tackled APT attack detection using network flow-based C&C detection method to detect the hidden C&C channel of unknown APT attacks and achieved an $f1 - score$ of 96.80% but did not provide the actual detection rate for their approach. Also, the author in [61] proposed a detection framework based on an enhanced SNN algorithm using semi-supervised learning approach on LANL dataset to scores suspicious APTs-related activities at three different stages of APT attack life cycle given a high weight rank to hosts depicting characteristics of data exfiltration with the believe that main APT attack is data exfiltration. This study faced a higher computational overhead cost.

In our previous work in [9], we proposed an approach using deep neural networks for APT multi-step detection which takes stacked LSTM-RNNs networks to automatically learn features from the raw data to capture the malicious patterns of APT activities using KDDCup99 dataset. This approach achieved a detection rate of 99.90%, see Table 3. This current paper proposed a framework named APT-DASAC based on stacked ensemble-LSTM variants, taken into consideration the distributed and multi-level nature of ICS architecture and reflect on the four main SCADA cyber attacks which are interception, interruption, modification and fabrication as recorded in [48] to demonstration the ability of this approach in detecting different stages of APT activities. This approach achieved an overall detection rate of 85% for NGP dataset and 93.67% for UNSW-NB15 dataset. Also, when ML-DT were implemented within our approach, we obtained 95% on both NGP and UNSW-NB15 datasets.

All the reviewed approach on this study has demonstrated a significant APT attack detection capability, however, none of these approach used the same dataset (see Table 3), making it difficult to rank the performance of these approaches. Also, the unavailability of a standard dataset or suitable public accessible dataset is a huge challenge in the field of cyber security, making it unfavourable to compare an APT detection system performance so as to chose an appropriate model for any given domain.

Table 3: Our Proposed Approach and other Works on APTs Detection

Proposed Method	Approach	Dataset	Outcome	Reference
Enhanced $SNN_{Algorithm}$	Semi-supervised learning approach	LANL	90.50%	[61]
BiLSTM&GCN	Network flow analysis	Malware Capture CTU-13 data warehouse	68.89% (APT IPs attack)	[46]
Network flow based on C&C detection method	DL techniques	Contagio blog malware	96.80% (f-score)	[47]
Stacked $RNN_{variants}$	DL techniques	KDDCup99	99.90%	[9]
APT-DASAC	ML - DT	NGP & UNSW-NB15	95%	This paper
APT-DASAC	Ensemble $LSTM_{variants}$	NGP & UNSW-NB15	85%	This paper

6 Conclusion

In this study, to overcome the issue of detecting APT dynamics attack life cycle, we have used supervised learning approach and a multi-layered attack detection

framework that takes into consideration the distributed and multi-level nature of ICS architecture and reflects on the four main SCADA-based cyber attacks. Therefore, a detection framework based on stacked-ensemble LSTM variants algorithm has been proposed and evaluated. This accounts as one of the contributions of this paper. Due to the dynamic nature of APT life cycle, APT attack cannot be detected automatically, hence this model serves as a supplement to automated IDS. The implemented algorithms achieved a competitive overall detection rate of 85%, 93.67% and 95% with micro-average ROC curve area of 91%. These results suggest that both stacked-ensemble LSTM variants and ML-DT approach are good candidates to be considered for developing an APT detection systems.

From Fig. 8, the value of $recall$ achieved also illustrates that when DL is used within the proposed approach, it did struggle to identify the relevant cases of command injection attack, DoS and Response Injection attacks within the NGP dataset. The class with more connection records seems to be learnt properly without confusing their identity while those with fewer connection records during training did not show good true positive rate as it was had to identify them. This indicates a data imbalance problem. However, this was not the case when ML was used in place of DL as the system achieved good $precision$ and $recall$ as evidenced in Table 3. Also, if the output from this study is compared to our previous work in [9], where we have implemented the same procedure with KDDCup99 dataset, the average detection rate achieved is 99.9% (see Table 3).

We can see that this approach performed very well on KDDCup99 dataset as the feature set contained within this data is highly distinguishable in nature. The result is slightly higher when both NGP and UNSW-NB15 dataset were used. This account as an identified issue from this study when it comes to comparing performance of various proposed detection framework with regards to accessibility and availability of suitable data / network flow information in security industries with respect to domain of interest.

Considering the different results obtained with three different datasets from diverse domains, our implemented approach showed a significant attack detection capability. This has also demonstrated that performance of attack detection approach applied can be influences by the nature of network connections with respect to the domain of application. This suggest that the ability and resilience of operational CPS state to withstand attack and maintain system performance are regulated by the safety and security measures in place, which is specific to that CPS devices or application domain. Hence, there is every need to investigation the nature of the network flow information within any system in mind to determine the security measures that will be suitable for that system.

References

1. Eke, H., Petrovski, A., & Ahriz, H. (2020). Handling minority class problem in threats detection based on heterogeneous ensemble learning approach. International Journal of Systems and

Software Security and Protection (IJSSSP), 11(2), 13-37.
2. Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., ... & Ueda, K. (2016). Cyber-physical systems in manufacturing. Cirp Annals, 65(2), 621-641.
3. Cardenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., & Sastry, S. (2009, July). Challenges for securing cyber physical systems. In Workshop on future directions in cyber-physical systems security (Vol. 5, No. 1).
4. Jazdi, N. (2014, May). Cyber physical systems in the context of Industry 4.0. In 2014 IEEE international conference on automation, quality and testing, robotics (pp. 1-4). IEEE.
5. Petrovski, A., Rattadilok, P., & Petrovski, S. (2015, September). Designing a context-aware cyber physical system for detecting security threats in motor vehicles. In Proceedings of the 8th International Conference on Security of Information and Networks (pp. 267-270).
6. Eke, H., Petrovski, A., & Ahriz, H. (2020). Detection of false command and response injection attacks for cyber physical systems security and resilience.. In13th International Conference on Security of Information and Networks (SIN 2020), November 4–7, 2020, Merkez, Turkey.ACM, NewYork, NY, USA, 8 pages. https://dl.acm.org/doi/10.1145/3433174.3433615
7. Kim, H. S., Lee, J. M., Park, T., & Kwon, W. H. (2000, November). Design of networks for distributed digital control systems in nuclear power plants. In Intl. Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human-Machine Interface Technologies (NPIC&HMIT 2000).
8. Humayed, A., Lin, J., Li, F., & Luo, B. (2017). Cyber-physical systems security—A survey. IEEE Internet of Things Journal, 4(6), 1802-1831.
9. Eke, H. N., Petrovski, A., & Ahriz, H. (2019, September). The use of machine learning algorithms for detecting advanced persistent threats. In Proceedings of the 12th International Conference on Security of Information and Networks (pp. 1-8).
10. Nissim, N., Cohen, A., Glezer, C., & Elovici, Y. (2015). Detection of malicious PDF files and directions for enhancements: A state-of-the art survey. Computers & Security, 48, 246-266.
11. Chandra, J. V., Challa, N., & Pasupuleti, S. K. (2016, March). A practical approach to E-mail spam filters to protect data from advanced persistent threat. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT) (pp. 1-5). IEEE.
12. Majdani, F., A., Batik, L., Petrovski, A., & Petrovski, S. (2020). Detecting Malicious Signal Manipulation in Smart Grids Using Intelligent Analysis of Contextual Data. ACM Digital Library: Proceedings of the 13 International Conference on Security of Information and Networks, 1-8.
13. NJCCIC. 2017. CRASHOVERRIDE NJCCIC Threat Profile, official site of the state of new jersey Original Release Date: 2017-08-10 and accessed on16/07/20. NJCCIC (2017).
14. Slowik, J. (2019). Evolution of ICS attacks and the prospects for future disruptive events. Threat Intelligence Centre Dragos Inc.
15. Linda, O., Vollmer, T., & Manic, M. (2009, June). Neural network based intrusion detection system for critical infrastructures. In 2009 international joint conference on neural networks (pp. 1827-1834). IEEE.
16. Harris, B., & Hunt, R. (1999). TCP/IP security threats and attack methods. Computer communications, 22(10), 885-897.
17. Giura, P., & Wang, W. (2012, December). A context-based detection framework for advanced persistent threats. In 2012 International Conference on Cyber Security (pp. 69-74). IEEE
18. Siddiqi, M. A., & Ghani, N. (2016). Critical analysis on advanced persistent threats. International Journal of Computer Applications, 141(13), 46-50.
19. Brand, M., Valli, C., & Woodward, A. (2010). Malware forensics: Discovery of the intent of deception. Journal of Digital Forensics, Security and Law, 5(4), 2.
20. Shashidhar, N., & Chen, L. (2011). A phishing model and its applications to evaluating phishing attacks.
21. McClure, S., Gupta, S., Dooley, C., Zaytsev, V., Chen, X. B., Kaspersky, K., ... & Permeh, R. (2010). Protecting your critical assets-lessons learned from operation aurora. Tech. Rep.
22. Alperovitch, D. (2011). Revealed: operation shady RAT (Vol. 3, p. 2011). McAfee.

23. Villeneuve, N., Bennett, J. T., Moran, N., Haq, T., Scott, M., & Geers, K. (2013). Operation" Ke3chang: Targeted Attacks Against Ministries of Foreign Affairs. FireEye, Incorporated. villeneuve2013operation
24. Singh, S., Sharma, P. K., Moon, S. Y., Moon, D., & Park, J. H. (2019). A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions. The Journal of Supercomputing, 75(8), 4543-4574.
25. Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research, 1(1), 80.
26. Odewale, A. (2018). Implementing secure architecture for industrial control systems. Proceedings of the 27th COREN Engineering Assembly, Abuja, Nigera, 6-8.
27. Nazarenko, A. A., & Safdar, G. A. (2019). Survey on security and privacy issues in cyber physical systems [J]. AIMS Electronics and Electrical Engineering, 3(2), 111-143.
28. Pasqualetti, F., Dorfler, F., & Bullo, F. (2015). Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems. IEEE Control Systems Magazine, 35(1), 110-127.
29. Cazorla, L., Alcaraz, C., & Lopez, J. (2016). Cyber stealth attacks in critical information infrastructures. IEEE Systems Journal, 12(2), 1778-1792.
30. Wu, G., Sun, J., & Chen, J. (2016). A survey on the security of cyber-physical systems. Control Theory and Technology, 14(1), 2-10.
31. Thakur, K., Ali, M. L., Jiang, N., & Qiu, M. (2016, April). Impact of cyber-attacks on critical infrastructure. In 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS) (pp. 183-186). IEEE.
32. Smiraus, M., & Jasek, R. (2011). Risks of advanced persistent threats and defense against them. Annals of DAAAM & Proceedings, 1589.
33. Lee, R. M., Assante, M. J., & Conway, T. (2017). CRASHOVERRIDE: Analysis of the threat to electric grid operations. Dragos Inc., March. https://www.dragos.com/wp-content/uploads/CrashOverride-01.pdf
34. Domović, R. (2017). Cyber-attacks as a Threat to Critical Infrastructure. INTEGRATING ICTIN SOCIETY, 259.
35. NJCCIC. (2017). CRASHOVERRIDE NJCCIC Threat Profile, official site of the state of new jersey Original Release Date: 2017-08-10 and accessed on 06/003/21. https://www.cyber.nj.gov/threat-center/threat-profiles/ics-malware-variants/crashoverride
36. Hemsley, K. E., & Fisher, E. (2018). History of industrial control system cyber incidents (No. INL/CON-18-44411-Rev002). Idaho National Lab.(INL), Idaho Falls, ID (United States).
37. Assante, M. J.,& Lee, R. M. (2015). The industrial control system cyber kill chain. SANS Institute InfoSec Reading Room, 1.
38. Carvalho, L. K., Wu, Y. C., Kwong, R., & Lafortune, S. (2018). Detection and mitigation of classes of attacks in supervisory control systems. Automatica, 97, 121-133.
39. Paoli, A., Sartini, M., & Lafortune, S. (2011). Active fault tolerant control of discrete event systems using online diagnostics. Automatica, 47(4), 639-649.
40. Bai, C. Z., Pasqualetti, F., & Gupta, V. (2017). Data-injection attacks in stochastic control systems: Detectability and performance tradeoffs. Automatica, 82, 251-260.
41. Miloševič, J., Tanaka, T., Sandberg, H.,& Johansson, K. H. (2017). Analysis and mitigation of bias injection attacks against a Kalman filter. IFAC-PapersOnLine, 50(1), 8393-8398.
42. Xu, Z., & Easwaran, A. (2020, April). A Game-Theoretic Approach to Secure Estimation and Control for Cyber-Physical Systems with a Digital Twin. In 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS) (pp. 20-29). IEEE.
43. Hu, G., Li, H., Xia, Y., & Luo, L. (2018). A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis. Computers in Industry, 100, 287-296.
44. Fawzi, H., Tabuada, P., & Diggavi, S. (2014). Secure estimation and control for cyber-physical systems under adversarial attacks. IEEE Transactions on Automatic control, 59(6), 1454-1467.

45. Shi, H., Xie, L., & Peng, L. (2021). Detection of false data injection attacks in smart grid based on a new dimensionality-reduction method. Computers & Electrical Engineering, 91, 107058.
46. Do Xuan, C., Nguyen, H. D., & Dao, M. H. APT attack detection based on flow network analysis techniques using deep learning. Journal of Intelligent & Fuzzy Systems, (Preprint), 1-17.
47. Shang, L., Guo, D., Ji, Y., & Li, Q. (2021). Discovering unknown advanced persistent threat using shared features mined by neural networks. Computer Networks, 107937.
48. Turnipseed, I. P. (2020). A new SCADA dataset for intrusion detection system research (Doctoral dissertation, Mississippi State University).
49. Morris, T., & Gao, W. (2014, March). Industrial control system traffic data sets for intrusion detection research. In International Conference on Critical Infrastructure Protection (pp. 65-78). Springer, Berlin, Heidelberg.
50. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In Eleventh annual conference of the international speech communication association.
51. Hagan, M. T., De Jesús, O., Schultz, R., Medsker, L.,& Jain, L. C. (1999). Training recurrent networks for filtering and control. Chapter, 12, 311-340.
52. Morris, T. H., Thornton, Z., & Turnipseed, I. (2015). Industrial control system simulation and data logging for intrusion detection system research. 7th annual southeastern cyber security summit, 3-4.
53. Sen, J. (Ed.). (2012). Cryptography and Security in Computing. BoD–Books on Demand.
54. Gao, W., Morris, T., Reaves, B., & Richey, D. (2010, October). On SCADA control system command and response injection and intrusion detection. In 2010 eCrime Researchers Summit (pp. 1-9). IEEE.
55. Gogoi, P., Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2012, August). Packet and flow based network intrusion dataset. In International Conference on Contemporary Computing (pp. 322-334). Springer, Berlin, Heidelberg.
56. Moustafa, N.,& Slay, J. (2015, November). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.
57. Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. International journal of pattern recognition and artificial intelligence, 23(04), 687-719.
58. Fawcett, T. (2006). An introduction to ROC analysis. Pattern recognition letters, 27(8), 861-874.
59. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications, 73, 220-239.
60. A. D. Kent, "Cybersecurity Data Sources for Dynamic Network Research," in Dynamic Networks in Cybersecurity, 2015.
61. Zimba, A., Chen, H., Wang, Z., & Chishimba, M. (2020). Modeling and detection of the multi-stages of Advanced Persistent Threats attacks based on semi-supervised learning and complex networks characteristics. Future Generation Computer Systems, 106, 501-517.

Index