Journal of Network and Systems Management (2021) 29:23
https://doi.org/10.1007/s10922-021-09589-6

AS-IDS: Anomaly and Signature Based IDS for the Internet of Things

Yazan Otoum[1] · **Amiya Nayak**[1]

Received: 29 September 2020 / Revised: 11 January 2021 / Accepted: 5 February 2021 /
Published online: 4 March 2021

Abstract

The Internet of Things (IoT) is a massively extensive environment that can manage many diverse applications. Security is critical due to potential malicious threats and the diversity of the connectivity. Devices can protect themselves and detect threats with the Intrusion Detection System (IDS). IDS typically uses one of two approaches: anomaly-based or signature-based. This paper proposes a model (known as "AS-IDS") that combines these two approaches to detect known and unknown attacks in IoT networks. The proposed model has three phases: traffic filtering, preprocessing and the hybrid IDS. In the first phase, the arrival traffic is filtered at the IoT gateway by matching packet features, after which the preprocessing phase applies a Target Encoder, Z-score and Discrete Hessian Eigenmap (DHE) to encode, normalize and eliminate redundancy, respectively. In the final phase, the hybrid IDS integrates signatures and anomalies. The signature-based IDS subsystem investigates packets with Lightweight Neural Network (LightNet), which uses Human Mental Search (HMS) for traffic clustering in the hidden layer and Boyer Moore is used to search for a particular signature in the output layer that is accelerated by using the Generalized Suffix Tree (GST) algorithm and by matching the signatures it classifies the attacks as intruder, normal or unknown. The anomaly-based IDS subsystem employs Deep Q-learning to identify unknown attacks, and uses Signal to Noise Ratio (SNR) and bandwidth to classify the attacks into five classes: Denial of Service (DoS), Probe, User-to-Root (U2R), Remote-to-Local (R2L), and normal traffic. Detected packets are then generated with new signatures, using the Position Aware Distribution Signature (PADS) algorithm. The proposed AS-IDS is implemented in real-time traffic with the NSL-KDD dataset, and the results are evaluated in terms of Detection Rate (DR), False Alarm Rate (FAR), Specificity, F-measure and computation time.

Keywords Internet of Things (IoT) security · Anomaly-based IDS · Signature-based IDS · Deep Q-learning · Lightweight Neural Network (LightNet)

✉ Yazan Otoum
yotoum@uottawa.ca

Extended author information available on the last page of the article

1 Introduction

IoT is used for many applications, including smart cities, industries and medical services. The applications deliver huge volumes of traffic to the end devices through the network, and since they deal with sensitive and non-sensitive data [1, 2], the attacks on the infrastructures are increasing. The detection of such attacks is a challenge, since there is minimal prediction efficiency for the various diverse attack methods. To counter this, an Intrusion Detection System (IDS) was developed to monitor and analyze network traffic, and make decisions regarding the network packets [3–5]. In the simplest case, the arriving traffic is analyzed by extracting the features in the packets, and differentiating normal from abnormal traffic using specific features, such as source IP address, destination IP address, source port number, destination port number and others. Security defense systems can detect different types of attacks, including DoS, Distributed DoS (DDoS) and spoofing. Each attack causes unique behaviour on the network, and the attack targets are different. In general, however, they utilize network resources and degrade channel characteristics.

The IDS is classified into two main types; signature-based and anomaly-based [6, 7]. The signature-based IDS stores a set of attack signatures, and identifies arrival attacks by validating their signature in the database. Thus, the use of signature-based IDS is efficient for identifying known attacks in the network, since the signatures are only those that were previously stored. The anomaly-based IDS can predict unknown attacks by monitoring and analyzing the traffic according to the packet features, which are capable of differentiating normal traffic from attack traffic. The challenges in these two IDSs follow:

1. *Signature-based IDS*: This type of IDS is suitable for detecting known attacks with signatures that are already in the database; the absence of signatures in the database allows all unknown attacks.
2. *Anomaly-based IDS* This IDS can detect even unknown attacks, but still requires an effective operating algorithm to correctly analyze the arrived packets.

Signature-based approaches have an advantage over the anomaly-based methods, as they are simple and can operate online in real-time. To ensure a more security aware environment, a hybrid IDS was designed that integrates signatures and anomalies [8–10]. The hybrid IDS incorporates machine learning algorithms such as a C5 decision tree, a Support Vector Machine (SVM), an OC-SVM, the k-Nearest Neighbor (k-NN) and others [11–13]. Deep learning algorithms are also used to help detect attacks [14], including Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). These deep learning algorithms improve detection efficiency. To test an IDS system that uses machine learning algorithms, a network dataset is collected and used to evaluate the system [15, 16]. This dataset is comprised of many packet features from network environments with network devices. The main challenges for IDS systems are:

– Improving the detection rate with accurate attack detection from the dataset using an efficient algorithm. The increase in the detection rate is critical, since it requires precise analysis of the packet features.
– Signature-based or anomaly-based can only be achieved by their method; either by detecting unknown attacks or by known attacks only.
– Slow processing of an algorithm to detect attacks degrades performance since the IoT has huge volumes of packets. The detection of abnormal packets needs to be faster and more efficient.

Challenges in IDSs are addressed in this work; the proposed hybrid IDS (AS-IDS) can operate quickly and efficiently to detect attacks, this is include reducing the detection time.

1.1 Motivation

IDS is a popular system used to detect network vulnerabilities, and the majority of IDSs apply machine learning algorithms that are efficient in detecting attacks [17, 18]. In general, IDSs are either signature-based, anomaly-based or a hybrid of the two. As discussed earlier, the limitations of the signature and anomaly methods is they can be overwhelmed when combined and used as a single system. The key motivation is to develop a hybrid IDS system that can predict any type of attack on a network. However, though using hybrid IDS with machine learning algorithms is efficient, it cannot effectively detect intruders when huge volumes of traffic enter the network, and it also lacks the ability to support real-time systems. Thus, for this purpose utilizing deep learning with IDSs is a promising solution [19, 20] that can achieve a higher detection rate than traditional machine learning methods. From the developed motivation in IDS, the objectives of the proposed model are as follows:

– To design a hybrid IDS framework that can overcome the issues in the signature and anomaly based IDSs,
– To reduce overhead during intrusion detection by the IoT gateway through a traffic filtering process,
– To reduce the feature dimensionality by using effective clustering processes before the intrusion detection, and
– To overcome the issues in the anomaly-based IDS by considering the environment related to the metrics.

1.2 Contributions

The contributions of the proposed AS-IDS system are summarized as follows:

– Initially, the gateways are used to filter arriving traffic by evaluating real network parameters, such as Signal-to-Noise Ratio (SNR) and bandwidth, while taking the packet features into account. As a result, some of the basic attackers are filtered out by the gateway directly.

– In AS-IDS, the signature-based IDS creates a repository to store signatures in the form of the Generalized Suffix Tree (GST). The arrived packets are then matched to the signatures using the Lightweight Neural Network (LightNet). In the LightNet sub-model, the Human Mental Search (HMS) and Boyer Moore algorithms are applied to group clusters and search the signatures. The mismatched signature holders are carried over to anomaly detection.
– In AS-IDS, the anomaly-based IDS investigates the packets using deep Q-learning, and learns from the environment via the estimated SNR and bandwidth constraints. In this way, it can predict unknown attacks by the analysis of packet features and abnormal channel metrics.

1.3 Paper Layout

The rest of the paper is organized as follows: Sect. 2 considers previous IDSs and their attack detection; Sect. 3 discusses the key problems in the IDS system when detecting attacks; Sect. 4 presents an effective solution with novel algorithms to solve the defined problems; Sect. 5 demonstrates the results of the proposed work and compares them to previous IDS; Sect. 6 summarizes the highlights of the proposed AS-IDS; and, Sect. 7 concludes the paper and summarizes potential future research directions.

2 Related Works

This section details the research conducted regarding different aspects of instructional detection systems with a variety of algorithms. In [21], an IDS was designed with the deployment of honeypots, and it was responsible for monitoring the network devices. It computed the belief, disbelief and uncertainty of each node's reputation from those that were managed by the honeypots. As a result, this IDS system was required to incorporate multiple honeypots in the IoT, which could be compromised and make the instruction detection more complex.

Shared model-based hybrid intrusion detection was performed on the IoT botnet dataset [22]. This involved four different phases to detect intrusions: preprocessing, feature selection and signature and anomaly-based IDSs. Features were selected using an information gain-based algorithm, then transmitted to the signature-based IDS model where a C5 classifier was applied. With the anomaly-based model, a one class SVM was used to detect intrusions. In this work, a one class SVM is unable to handle immense dimensional datasets, as it tends to decrease the detection rate. Then, for the newly detected attacks it generates an attack signature for further processes. In [23], a combination signature and anomaly-based IDS was proposed that used Random Forest (RF) and Boruta algorithms. In the work, the RF computes the Z-score value for each feature using entropy and the Gini index, and then the attacks are classified according to the default RF parameters. The use of RF was based on the tree, which occupies large memory and cannot process classification results quickly.

In [24], the authors proposed a hybrid multi-model solution that utilizes an Ensemble model with stacked generalization. In this work, Random Forest (RF), Logistic Regression (LR) and k-NN are used for training purposes, and SVM based-Stacking was applied for testing. As a result, the SVM classifier algorithm is unable to process effectively when the dataset is in IoT systems, as they have a vast volume of traffic which renders SVM unsuitable. In [25], k-NN and K-means clustering algorithms were proposed for a hybrid IDS. Initially, the pre-processing process for the dataset was performed and then K-means and k-NN were used for clustering and classification, respectively. Normal and attack data types were clustered based on the computed centroid and the distance between points, and from the clustered data k-NN was applied for classification. However, in K-means the selection of the K-value is critical, since if it fails then the clustering process will be inefficient.

A misused detection-based model scheme (Signature-based) was used in [26]. In their work, KDD'99 and UNSW-NB15 datasets were used to detect intrusion, based on the attack signatures. The authors used the kernel principal component analysis algorithm to reduce features' dimensionality and extract significant features, while the Extreme Learning Machine (ELM) algorithm was applied for intrusion detection. This had hybrid kernel functions such as Radial Basis Function (RBF) and polynomial kernel for detection, and the parameters of the ELM were optimized using the Differential Evolution (DE) algorithm with the Gravitational Search Algorithm (GSA). The main drawback of this research was that the Kernel Principal Component Analysis (KPCA) was used to reduce the features' dimensionality by selecting optimal features before the intrusion detection process. Thus, the results indicate low accuracy during the intrusion detection, since the size of the kernel matrix increases quadratically as the dataset size increases.

The optimization algorithms were also used in the intrusion detection systems, and as in [27], intrusion attacks are detected using the features selection-based algorithms. The Pigeon Inspired Optimization (PIO) algorithm and continuous binary cosine models were applied to select the significant features, and to detect attacks from the dataset. Optimization with AdaBoost machine learning algorithms to detect network intrusions were proposed in [28]. The model first preprocesses the data packet, then performs feature selection using the Artificial Bee Colony (ABC) optimization algorithm. This requires frequent estimates of the fitness values, which could lead to increased processing time. In [29], three different processes were proposed: preprocessing, feature selection and classification. Preprocessing was done by converting the data in the dataset into respective numeric values, and Principal Component Analysis (PCA) was used to select the optimal features and the Genetic Algorithm (GA) based Deep Belief Network (DBN) classifier was applied. The GA was used to select the optimal parameters for the DBN algorithm, though it takes more time for optimal selection.

In [30], the authors proposed a multi-objective optimization process to minimize the rates of false positives and negatives through detecting a group of generated alerts from various IDSs. The model has four phases; In the first phase, the low-level alerts are classified into meta-alerts for each IDS. Then, the featured meta-alerts are filtered into a set of P-FNs. Next, a clustering step inter-IDS is performed to groups similar meta-alerts together to avoid redundancy. In the last phase, a binary

multi-objective optimization problem (BMOP) is used to detect FNs and FPs. The proposed model is evaluated using real network traffic, NSL-KDD, and DARPA 1999 datasets. Experimental results show that the proposed process detects up to 98.8% of false negatives and positives.

Deep learning algorithms are also proposed for the IDS, such as that in [31]. Compared to machine learning, deep learning was able to manage large volumes of data, and is thus most suitable for large datasets. In [32], due to extensive traffic from an IoT an anomaly-based IDS that uses a deep learning algorithm was applied. It began by taking packet features such as the IP address and reception rate into account. The intrusion detection process used a Deep Belief Network (DBN) with a feed-forward Deep Neural Network (DNN). Hence, deep learning is a promising solution to identify attacks. In [33], the authors used preprocessing, feature extraction and classification phases. With preprocessing, the normalization process was applied and the significant features were extracted from the dataset. These features were then processed in the DBN layer, which is also used for signature verification. The process of matching the signatures of incoming packets with the database signature results is very tedious, plus it degrades attack detection accuracy and takes longer.

An anomaly-based IDS model in which anomalous and normal packets are classified was proposed in [34]. Four different classifiers were used to detect the anomaly-based IDS: SVM, Decision Tree (DT), Random Forest (RF) and Gradient Boosting Tree (GBT). The misused detection model operated using the Convolutional Long Short-term Memory (ConvLSTM) algorithm, with the features extracted by applying the convolutional algorithm then classified by the LSTM algorithm. The incoming packets are first processed in the anomaly-based IDS model, where high feature dimensionality of the incoming packet streams is not reduced. This causes high computation times and degrades the detection rate. The authors in [35] have adopted a three-phase deep learning-based model to secure the IoT. The Minkowski distance method is applied to remove redundant data from the NSL-KDD dataset, and the Spider Monkey Optimizer (SMO) algorithm is used to select the optimal features from the dataset, which is then further processed in the Stacked Deep Polynomial Network (SDPN) to detect network intruders. From previous literature, the use of deep learning, machine learning and optimization play a vital role in the detection of intrusions, though they all have critical issues that degrade performance. Table 1 shows a comparison of the previously mentioned works.

3 Problem Description

A two layer mechanism was proposed to detect intrusions in IoT networks, using Principal Component Analysis (PCA) and Latent Dirichlet Allocation (LDA) to reduce the dimensionality of the features [36]. In addition, the reduced feature set was further processed in the Naïve Bayes and k-NN algorithms, also to detect intrusions. Naïve Bayes then classifies the normal, anomalous and k-NN for intrusion into the IoT. The main significant issues in the work are:

Table 1 Comparison of related works

Model	Dataset	IDS approach	Algorithms	Application
Ensemble HIDS [22]	Bot-IoT	Signature and anomaly-based intrusion detection system	C5 and SVM	IoT
Improving IDS by estimating parameters of RF in Boruta [23]	NSL-KDD dataset	Signature and anomaly-based intrusion detection system	Random Forest (RF) and Boruta Algorithms	IoT
A stacking ensemble for network intrusion detection [24]	UNSW NB-15 and UGR'16	Signature-based intrusion detection system	Random Forest (RF), logistic regression, K Nearest Neighbor (KNN), and Support Vector Machine (SVM)	General computer networks
Hybrid IDS using K-means and Random Tree [25]	KDD'99 dataset	Signature-based intrusion detection system	K-means and Random Tree algorithms	General computer networks
Novel IDS based on an optimal hybrid kernel extreme learning machine [26]	UNSW-NB15 and KDD'99 datasets	Signature-based intrusion detection system	Gravitational Search Algorithm (GSA), Differential Evolution (DE), and Principal Component Analysis (KPCA) algorithms	General computer networks
A feature selection algorithm for IDS-based on PIO [27]	KDD'99, NLS-KDD and UNSW-NB15 datasets	Signature-based intrusion detection system	Pigeon Inspired Optimizer (PIO)	General computer networks
Anomaly network-based IDS using a reliable hybrid artificial bee colony and AdaBoost algorithms [28]	NSL-KDD and ISCXIDS2012 datasets	Anomaly-based intrusion detection system	Artificial Bee Colony (ABC) and AdaBoost algorithms	General computer networks
Intrusion detection for IoT based on improved genetic algorithm and deep belief network [29]	NSL-KDD dataset	Signature-based intrusion detection system	Genetic algorithm (GA) and deep belief network (DBN) algorithms	IoT
Enhancing the accuracy of IDSs by reducing the rates of false positives and false negatives through multi-objective optimization [30]	DARPA 1999 and NSL-KDD datasets	Signature-based intrusion detection system	Multi-Objective Optimization Problem (BMOP)	General computer networks

Table 1 (continued)

Model	Dataset	IDS approach	Algorithms	Application
Towards deep-learning-driven intrusion detection for the IoT [32]	Captured IoT network-traffic	Anomaly-based intrusion detection system	Deep belief network (DBN) and deep neural network (DNN)	IoT
Deep belief network enhanced ids to prevent security Breach in the IoT [33]	IoT network-traffic	Signature-based intrusion detection system	Deep belief network (DBN)	IoT
A scalable and hybrid ids-based on the convolutional-LSTM network [34]	ISCX-UNB dataset	Anomaly and misused-based intrusion detection system	Convolutional-LSTM network	IoT

- The proposed concept is based on a layer-based dimension reduction and classification model, which is a tedious computational process for intrusion detection in an IoT device. It results in high computational time during intrusion detection because it initially reduces the feature dimensions using two different algorithms, then the classification is also done using two different algorithms.
- Using classifiers such as Naïve Bayes and k-NN does not provide accurate intrusion detection results. Naïve Bayes results are based on the probability of intruders, while k-NN is unable to manage the outliers of the intruder datasets.
- The paper achieves a low detection rate due to ineffective feature reduction procedures during intrusion detection. The proposed PCA lacks significant features due to its ineffective principal components (multi-dimensional mean, square distance of features) selection procedures.

Optimization with deep learning algorithms has also been used in intrusion detection systems, and in this work [29] the GA was improved using elite retention strategy. The improved GA was applied to select the optimal number of hidden layers and the number of neurons in each layer in the DBN, and the DBN was then used to detect intrusions in the IoT. Though DBN performs well, this work revealed the following issues:

- The proposed intrusion detection framework doesn't provide an accurate detection rate since it involves processing all features (41) in the dataset, which also contains irrelevant features such as duration, land and hot. As the processing of these features consumes extra time, this work was unable to detect intruders over an acceptable period.
- Here, the proposed DBN does not produce optimal intrusion detection results, since its learner has issues of prematurely converging. Thus, this work failed to detect intruders accurately, which reduced the intrusion detection rate.

In [37], a hybrid IDS was proposed for signature-based misuse detection and anomaly-based detection. The signatures in the repository were constructed in a tree format with the suffix tree algorithm. For the received packets, the signature-based repository pattern matching method was applied to detect attacks, and for the unknown signature patterns the received packets were transmitted to the anomaly detection engine, and RNN was used to detect intrusions. The key problems with this hybrid IDS are:

- The dataset used in this work to detect intrusions is in raw form, and has not eliminated redundancies. However, it does contain irrelevant and redundant features, and addressing these in intrusion detection results in poor accuracy.
- Here, the signatures are stored in the form of a tree using a suffix tree algorithm. This does not provide better performance in high dimensional data environments however, since it constructs suffixes for a single string rather than a set of strings.

The main problems defined in this section are solved by the proposed AS-IDS, using signature and feature analyzes of arrived traffic. Since IoT deals with great number

of devices, it is essential to develop a hybrid IDS able to adapt to the unique behaviour of each device.

4 The Proposed Model

The proposed AS-IDS model combines signature-based and anomaly-based IDSs, Which can be detected both known and unknown attacks. The model is comprised of three phases; traffic filtering, preprocessing and the hybrid IDS phase.

Figure 1 illustrates the proposed AS-IDS model, and shows the used algorithms. IoT gateway has the capability to perform the filtration for the arriving traffic through verifying the main traffic parameters and filtering out the mismatched packets. This process is applied to the real-time traffic by matching the dataset features, and the benchmark dataset enters to perform preprocessing and training in the hybrid IDS phase. In the preprocessing phase, the dataset features are decoded and normalized and redundancies are removed. After preprocessing, the dataset enters the hybrid IDS phase that integrates signatures and anomalies. In hybrid IDS, the signature is matched in LightNet using an HMS algorithm from the constructed signatures tree. All known attacks are identified by the signature-based IDS analyses, and unknown attacks are identified by an anomaly-based IDS using a Deep Q-learning algorithm that can learn from the environment. Due to reinforcement learning, if a new packet arrives it can still be predicted by the IDS. Hence, the proposed AS-IDS is efficient,

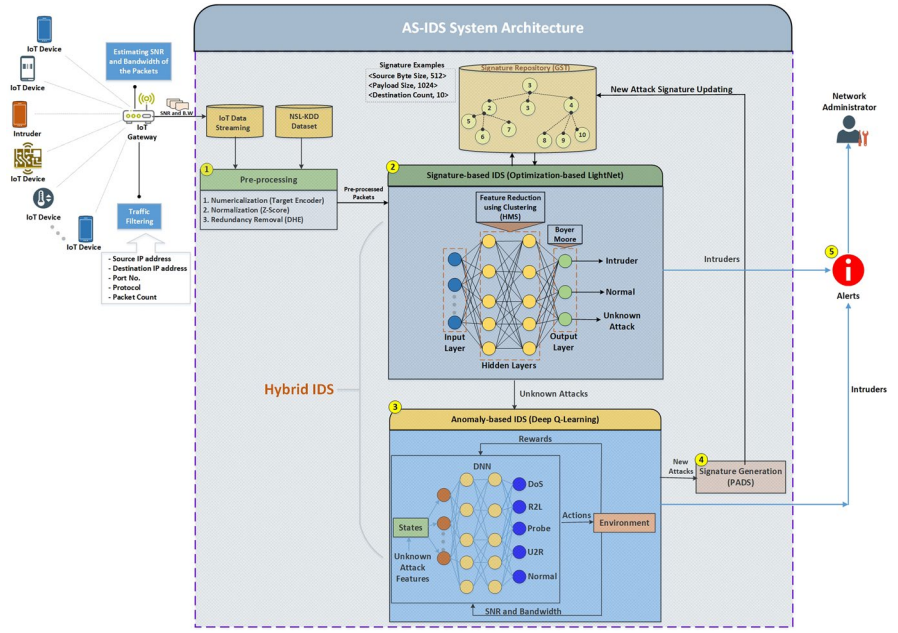

Fig. 1 Proposed AS-IDS Model

and the use of deep learning ensures optimal performance when high volumes of traffic enter the system.

4.1 Traffic Filtering and Preprocessing

The first phase in the proposed system is traffic filtration of the arriving traffic. The process of filtration is handled by validating the source IP address, destination IP address, port numbers, protocol and packet count. Using these features, the arrived traffic is matched in the gateway, where abnormal features are filtered out and basic attacks are blocked. For each unfiltered packet passed through the IoT gateway, the SNR and bandwidth are computed by the gateway. Assuming an IoT device is IoT_i and it submits the packet $\mathrm{P}_i = \{\mathrm{S}_{IP}, \mathrm{D}_{IP}, \mathrm{P}_{no}, \mathrm{P}_t, \mathrm{P}_{ct}\}$ which represent the source IP address, destination IP address, port number, protocol and packet count, respectively. These packet features are cross-verified by the gateway, then move on to preprocessing. The packet filtering pseudo-code is as follows:

Algorithm 1: Packet Filtration

```
Begin;
For all (IoT_1,...,IoT_i);
if P_i (S_IP,D_IP,P_no,P_t,P_ct )= Predefined Values then
 |  Allow Packet P_i from IoT_i
else
 |  Discard Packet
end
End
```

In the second phase, the preprocessing of the dataset starts by encoding the string values in the dataset into numeric values using the Target Encoder algorithm; for example, the protocol type field has TCP, UPD or ICMP string values. The Target Encoder algorithm groups the data by category, tracks the number of occurrences of each target, and calculates the probability of each Target, based on the computed mean value. The target fields in the dataset are converted into numeric values. After conversion to numeric values, a Z-score is used to normalize the dataset. The Z-score is calculated by the following mathematical formula:

$$Z = (X - \mu)/\ \sigma \tag{1}$$

Here X, μ and σ represent the original feature vector, observed mean and standard deviation values, respectively. By using this simple normalization the dataset fields reach the range of [0-1], which will improve the classification process. To remove redundancies in the dataset which can increase processing time and degrade classification performance results, we adapted the DHE [38], that uses H-functions. Thus, the Hessian matrix, coordinate function and constant function identify the similarities between values, reduce the dimensionality by determining the local neighbours and compute the tangent coordinates. In this way, the dataset dimensionality is

reduced in the preprocessing phase, and attacks are detected by analyzing the packet features.

4.2 The Hybrid IDS Phase

The hybrid IDS subsystem has two main processing sections that combine signature-based and anomaly-based IDS. The signature-based IDS is performed first to detect all known attacks by matching the stored signatures. The signatures are generated from Position Aware Distribution Signature (PADS) algorithm [39]. The signature is maintained in the repository using a Generalized Suffix Tree (GST), which can match signatures in an asymptotically optimal time.

Let L and M be two signatures the suffix tree is built for. The new signature is generated as L#M$, where # and $ represent the suffixes of L and M. If the size of the signatures are m it uses O(m+n) to match the signature repository. The LightNet algorithm [40] is used to detect known attacks in the signature-based IDS submodel. HMS [41] is applied to cluster at the hidden layer, and the Boyer Moore algorithm searches the output layer. The LightNet is designed from continuous weight networks. Most of the weight values are 0, and non-zero weights are limited to two either -1 or +1. This algorithm follows the synaptic pruning training process, and the activation function in LightNet is represented by odd or hyperbolic tangent expressions. Thus, the arbitrary location is defined as:

$$\tanh(x-\rho) + \tanh(-(x-\rho)+\chi) \tag{2}$$

where, $\rho, \chi \epsilon \mathbb{R}$.

LightNet is comprised of three layers: input, hidden and output. Packet features are considered to be input, and the hidden layer HMS is used to cluster similar packet features. This HMS algorithm proposes two main processes: searching by Levy flight or by grouping. The Levy flight process is performed based on the following Levy Distribution expression:

$$L(x) = \frac{1}{\pi}\int_0^{\infty} \exp\left(-\alpha q^{\beta}\right)\cos(qx)dx \tag{3}$$

Here, α denotes the scaling factor, β is the distribution index that is limited to $0 < \beta \leq 2$. Then, the generation of step size is given as:

$$S = \left(2 - itr * \left(\frac{2}{itr}\right)\right) * \alpha \bigoplus L(x) \tag{4}$$

where *itr* represents the number of iterations, and the product $\oplus$ means entry-wise multiplications.

The similar signatures of the dataset are clustered with the clustering K-means algorithm, then the k-value is determined and the signatures are clustered in the hidden layer. After this, the output layer is responsible for searching using the Boyer Moore pattern matching algorithm, which is known as an effective suffix heuristic process since it conducts bad character and good suffix approaches. The

bad character, is the character of the text which doesn't match with the current character of the pattern. based on this mismatched, the pattern is shifted until the mismatch becomes match or the pattern pass the mismatched character. With the good suffix approach, the signature string matches the pattern by the following four steps:

Step 1 Signature S in pattern P matches at time t;
Step 2 Pattern P with the prefix matches the suffix;
Step 3 The P moves all the characters to S; and,
Step 4 It generates match or mismatch results in S.

The hybrid IDS phase applies a signature-based IDS that detects known attacks by matching signatures in the tree. Using LightNet, the received packets are classified into three classes: intruder, normal and unknown attack. The intruder packets are reported, and the unknown packets are analyzed in the anomaly-based IDS to precisely identify the attack type.

The classification of the signature-based IDS is then carried out by an anomaly-based IDS, and only the unknown attacks are processed. In the anomaly-based IDS, a deep Q-learning algorithm that considers SNR and bandwidth parameters classifies the attacks as DoS, Probe, User-to-Root (U2R) or Remote-to-Local (R2L). With Q-learning, the environment is learned by the agents and generates a Q-table matrix that has the states (Features) and the actions (Send/Don't Send Alert). However, Q-learning is only suitable for small scale environments, and the IoT is an exceedingly large-scale environment. Thus, Q-learning is combined with deep learning to become a Deep Q-learning algorithm that can process multiple unknown attack packets simultaneously.

Each packet consists of SNR and bandwidth as input to the input layer of the Deep Q-learning algorithm. The benefit of using reinforcement learning is it evaluates the previous result, and can determine optimal future actions, while deep learning is the optimal classifier algorithm and is suitable for large volumes of inputs. The combination of these allows Deep Q-learning to apply the classification.

Let the states and actions be represented as $(S_1, S_2, S_3, \ldots S_t)$ and $(A_1, A_2, A_3, \ldots A_t)$ respectively. The Q-value in this Deep Q-learning is determined using the following expression:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \max_{\alpha} Q(S_{t+1}, a) - Q(S_t, A_t)\right] \tag{5}$$

where R_{t+1} is defined as a reward by 1 on each timestep, according to the attack detection decision. Thus, the learning agent in this reinforcement learning algorithm learns policy $\pi(A_t|S_t)$, If γ is the learning rate, S_t and A_t are the state and action for that specific packet, respectively. An epsilon-greedy policy is applied in deep Q-learning to perform the actions. According to the proposed algorithm, the states depend on the SNR and bandwidth of the packet and other significant packet features to detect four different attacks (i.e. DoS, Probe, U2R, R2L) that are unspecified in the signature-based IDS.

Table 2 details the states and actions that are defined and, due to agents' ability to learn, new states and actions are defined that make future predictions more accurate. The nodes in Deep Q-learning know which previous decision-making experience is key to improving future decisions. Thus, the loss function is predicted from the mean square error Q-value, as well as the target Q-value. Based on deviations of the major elements of the packet features, attacks are differentiated and detected.

Algorithm 2: Anomaly-based IDS

```
Begin;
For all (S_t,A_t);
if S' is terminal then
    Compute new initial state from the reward
else
    S←S';
    Return attack type {DoS,Probe,U2R,R2L}
end
End
```

For detected attacks, the signature is generated and updated in the repository tree, as it is essential to eliminate attacks in the signature-based IDS when it occurs in the future. This is done using signature generation PADS, which contain segments of both anomalous and standard signatures.

The byte frequency distribution of the traffic is computed and compared with the distribution of normal traffic [39]. A large difference is considered as an anomalous. Anomalous signature positions the signature length w with respect to the byte frequency distribution, where W is the width of the signature in terms of the number of bytes.

After the signature is generated, it is updated in the repository which is maintained as a tree. From the anomaly-based IDS it is classified as Dos, Probe, U2R, R2L or normal. Thus, the proposed AS-IDS model ensures efficient attack prediction from the packets, as well as the network parameters which are also important for intruder detection.

Figure 2 depicts the complete workflow of the proposed AS-IDS model that integrates signature-based and anomaly-based IDS. As indicated by the flow of

Table 2 State-Action Pairs

State	Action	New (State,Action)
$\mathbf{S_1}$	A_1	S'_1, A'_1
$\mathbf{S_2}$	A_2	S'_2, A'_2
$\mathbf{S_3}$	A_3	S'_3, A'_3
$\mathbf{S_4}$	A_4	S'_4, A'_4
⋮	⋮	⋮
$\mathbf{S_t}$	A_t	S'_t, A'_t

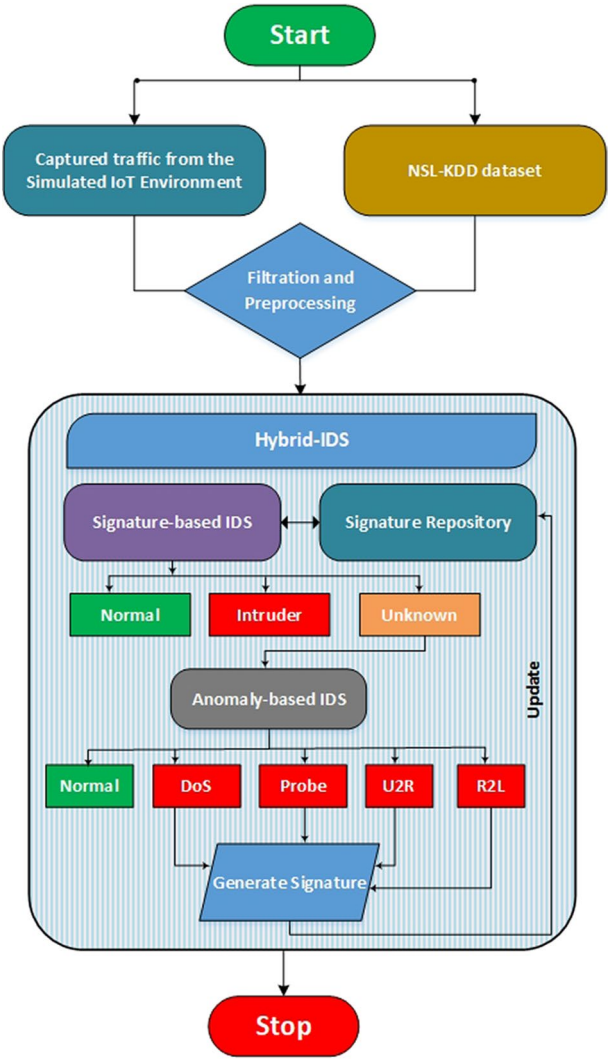

Fig. 2 Overall workflow of AS-IDS

the proposed model, the IDS system begins by collecting the traffic from the IoT devices, and analyzes it in the IDS to predict the behaviour of packets with respect to the network and packet features. With the proposed model, the classified intruders are detected and alert messages are sent to the network administrator regarding the involvement of intruder traffic in the network. From this evaluation, the network administrator can manage network intruder issues sooner. A network with numerous possible intruders will decrease network performance in terms of limited resource utilization, longer transmission times, wasted channels

and others factors. Thus, detection of intruders and attackers by the network dataset will help increasing the network performance.

5 Experimental Evaluation

In this section, the experimental evaluation of the proposed model is compared with other algorithms. This includes simulation setup, dataset description, comparative analysis and highlights. The efficiency of the proposed model is determined based on the comparisons.

5.1 Simulation Setup

The AS-IDS is developed by a network simulator and an IDS dataset that can determine intruder behaviour in the system. The NS3.26-based network simulation is performed on packets from the IoT nodes that are designed to behave as real nodes in an actual network environment. Table 3 shows implementation parameters for the simulated network.

NS3 for IDS incorporates the proposed algorithms into the system to detect intruders. All the algorithms are written in C++, and called by Python script. Based on the C++ algorithms, the results are evaluated as graphic plots of the significant performance metrics.

Figure 3 shows some screenshots of the implemented environment using NS-3. Where the simulated 50 IoT nodes are shown in both NS-3 PyViz, which is a real-time simulation visualizer that used for debugging purposes without needing a trace file, and the NetAnim offline animator that uses XML trace files collected during the simulation to show the network topology and animate the packets flow between nodes.

5.2 Dataset Description

In this proposed model, NSL-KDD dataset is used which is an enhanced version of the KDDCUP'99 dataset. This dataset composes of training and testing dataset with 125973 and 22544 records.

Each NSL-KDD record has 41 features (e.g., protocol type, Logged in, and Duration). These features are represented as numeric, nominal, and binary, defined as continuous or discrete, and labelled as normal or attack. Table 4 describes the NSL-KDD attacks types and the number of attack records in the training and testing sets is depicted in Table 5. The training and testing sets contain a total of 22 and 17 attack types, respectively.

Table 3 Implementation parameters

Parameter	Range
Simulation Specification	
Network area	250 × 250 m
Number of IoT nodes	50
Number of IoT gateway	1
IDS	1
Mobility Model	Randomwaypoint
Signature based IDS classes	Normal, Intruder, Unknown
Anomaly based IDS classes	Normal, Dos, Probe, U2R, R2L
Number of packets	100
Packet interval	1 s
Packet size	512
Simulation time	300 s
System Specification	
Simulator version	NS3.26
Operating system	Ubuntu-14.04 LTS
System type	32-bit
Processor speed	2.5 GHz
System processor	Intel Core i7

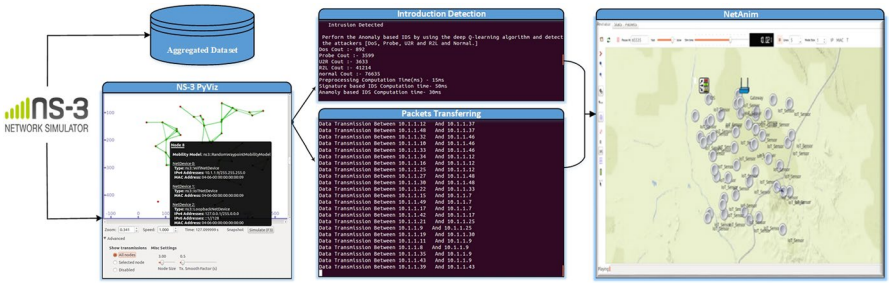

Fig. 3 Implementation environment

5.3 Comparative Analysis

This section highlights the metrics of the significant constraints of the proposed model, and compares them with Deep Belief Network (DBN) and Deep Recurrent Neural Network (DRNN) algorithms that are used previously in the hybrid IDSs [29, 37, 42]. The main performance metrics used are the detection rate, false alarm rate, sensitivity, specificity and F-measure.

Table 4 Attacks in NSL-KDD dataset

Attack type	Description
DoS	It is a type of flooding data packets that occupies larger resource in the network
Probe	This attack is defined based on the information gathering i.e. it collects data from the other nodes
U2R	This attack defines the involvement of access requests from unauthorized root node or super user
R2L	This attack performs on local access of unauthorized nodes using devices present in remote locations

5.4 Detection Rate and False Alarm Rate

Detection Rate (DR) is defined as the calculated ratio between the numbers of correctly detected event e.g. attack, and the total number of these events. as shown in the following equation:

$$DR = \frac{TP}{TP + FN} \tag{6}$$

where, TP, FN are True Positive, and False Negative respectively. While, False Alarm Rate (FAR) is defined as the calculated ratio between the number of negative events e.g. attacks that are incorrectly detected as positive (false positives) and the total number of the actual negative events and expressed as the following equation;

$$FAR = \frac{FP}{FP + TN} \tag{7}$$

The more effective detection system has a higher detection rate and lower false alarm rate.

Figures 4 and 5 show the detection and false alarm rates of deep learning approaches and the proposed AS-IDS.

The proposed AS-IDS has a higher detection rate due to the use of both signature-based and anomaly-based detection to identify attacks. The matching of signatures is also more accurate due to the use of the Boyer Moore method and anomaly detection by deep learning, as well as the use of environmental parameters that help improve the detection rate.

Table 5 NSL-KDD records details

Attack type	Number of records	
	Training set	Testing set
DoS	45927	7456
Probe	11656	2421
U2R	52	200
R2L	995	2756

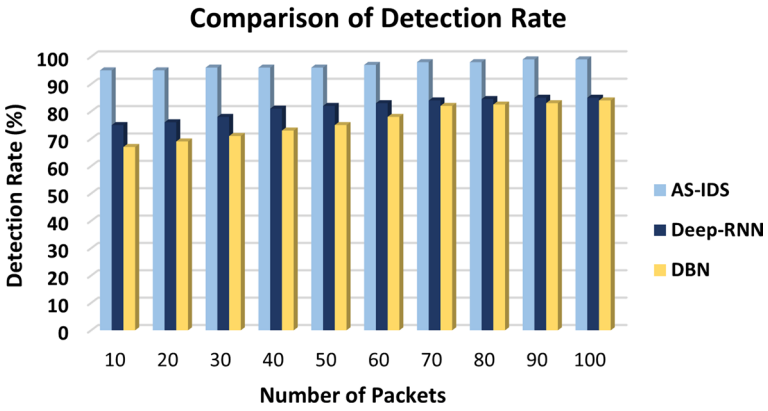

Fig. 4 Detection rate comparison

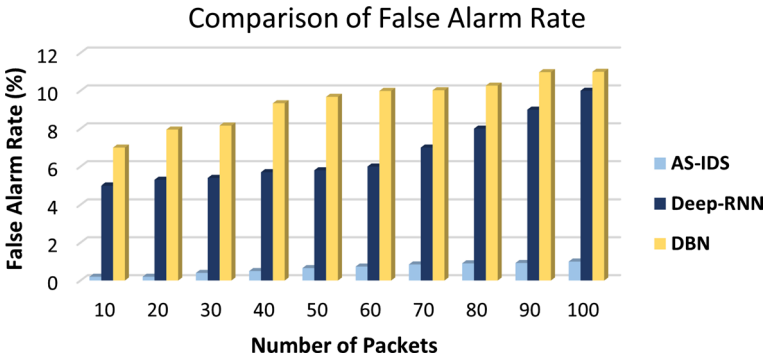

Fig. 5 False alarm rate comparison

On average, the detection rate of AS-IDS is 96.9%, which ensures support for the continuous increase of packets in the network. DBN and Deep-RNN algorithms have lower detection rates of 76.4% and 81.3%, respectively. This is due to the lack consideration of the significant features used for the classification, and emphasizes the importance of the significant features in the IDS systems, even with deep learning algorithms. The improvement in detection rates will reflect the decrease of false alarm rates in the proposed model. The false alarm rate is defined as the number of attacks that are not correctly detected by the IDS. They are reported to the administrator as an attacks on the network.

A higher false alarm rate indicates that the model performs poorly in identifying attacks. The reasons for a decrease in detection rate and an increase in false alarm rate are follows:

– The dataset of traffic is collected and used raw, which can introduce redundancy and degrade classification results due to the need to correlate the normal and redundant data. In addition, the processing of redundant data requires more time since the dataset is larger. Processing IDS using a dataset will always create redundant data, and processing with the redundant data will degrade the performance of the system significantly.
– Other existing works may fail to take significant features into account. This can cause attack packet behaviour to be detected from the features less efficiently, since each attack packet has different features. To increase the detection rate, it is essential to consider most significant packet features for processing.
– Although deep learning algorithms can learn the features dynamically through the training data process, they can't learn the current environment parameters as when using the reinforcement learning.

Considering the supporting data, the proposed model performs better than existing works. Therefore, the proposed AS-IDS has higher performance than the Deep-RNN and DBN algorithms when using with the IDSs.

5.5 Sensitivity, Specificity and F-measure

Sensitivity and specificity parameters play a vital role in the evaluation of IDS performance that classifies attacks. The sensitivity defines a true positive rate, and the specificity defines a true negative rate. The sensitivity is computed based on the proportion of positive classes made up of attackers and non-attackers. In turn, specificity is computed from the proportion of detected negative attacks from the dataset.

Sensitivity and Specificity performance with respect to increases in packets are evaluated, and the results are shown in Figures 6 and 7. From the investigations of AS-IDS, Deep-RNN and DBN, the proposed AS-IDS shows improvement regardless of the number of arrival packets. The two parameters are computed by mathematical expressions based on the classification results. The sensitivity and specificity are given as follows:

$$Sensitivity = \frac{N(TP)}{N(TP) + N(FN)} \tag{8}$$

$$Specificity = \frac{N(TN)}{N(TN) + N(FP)} \tag{9}$$

where N(TP), N(TN), N(FP) and N(FN) denote the number of true positives, true negatives, false positives and false negatives, respectively. The higher sensitivity and specificity indicate that the proposed system has a better performance than other algorithms. Based on this, the sensitivity represents the precision of the prediction of normal packets, and the specificity identifies the correctness in the classification of the attacks packets. The overall performance in terms of sensitivity is 96.6% for the proposed AS-IDS. 76.4% and 80.3% in DBN and Deep-RNN classifiers for IDS, respectively. The differences of 22% and 16.3%, means that the proposed AS-IDS

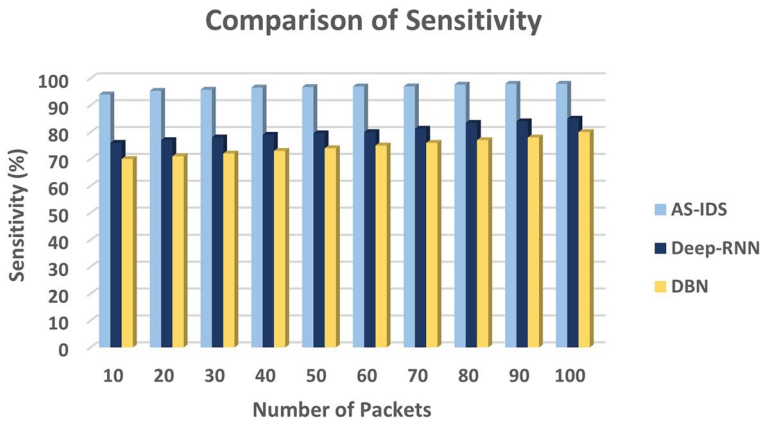

Fig. 6 Sensitivity comparison

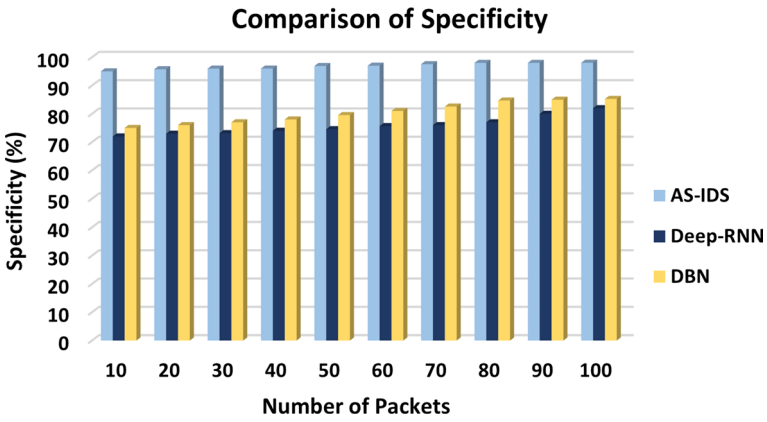

Fig. 7 Specificity comparison

functions are superior to other deep learning methods. This is due to the preference tendency of significant features, and improved layer processing in LightNet. In addition, preprocessing indicates improved sensitivity. Similarly, the specificity results show growth with respect to increasing numbers of packets. This increase in specificity is 96.8% for the proposed model, and 75.7% and 80.4% in Deep-RNN and DBN, respectively. This evaluation indicates that the proposed AS-IDS results have higher detection performance than other IDSs.

Accuracy of classification results is achieved by estimating the F-measure parameter, which is determined from the true positive, true negative, false positive and false negative values in the classification. Figure 8 illustrates the performance of the proposed AS-IDS compared to other deep learning-based IDS.

5.6 Execution Time

Execution time is defined as the period required for a model to perform a specific task in order to deliver a particular output. The execution time should not be overly high with increased numbers of inputs, and it should not degrade the performance of the system. Figure 9 illustrates execution times of the IDS system for preprocessing, signature computation, and anomaly computation.

With AS-IDS, signature and anomaly-based IDSs are performed. The comparison shows high deviations between the proposed model and the other algorithms for detecting the attacks. Deep-RNN requires more testing time, as it retains a memory of the previous results in the hidden layer nodes. Overall, the related classification parameters improved, and the processing time in AS-IDS was lower. Hence, the proposed AS-IDS system can detect attacks efficiently, and process large volumes of arriving traffic.

6 Proposed AS-IDS Research Highlights

Reducing high false alarm and false positives rates remain challenging issues for intrusion detection in IoT environments. None of the works in the literature have focused on managing high stream packets in IDS perception layers, and most hybrid and signature based-intrusion detection models are based on pattern matching algorithms. However, these methods can only work under single packet verification, not HTTP traffic-based environments (IoT). A major issue with anomaly-based IDS is inefficient linking of abnormal and intrusive factors, and none of the works have concentrated on abnormal inducing factors such as SNR and bandwidth. In addressing the defined problem statement, this proposed work developed the following highlights:

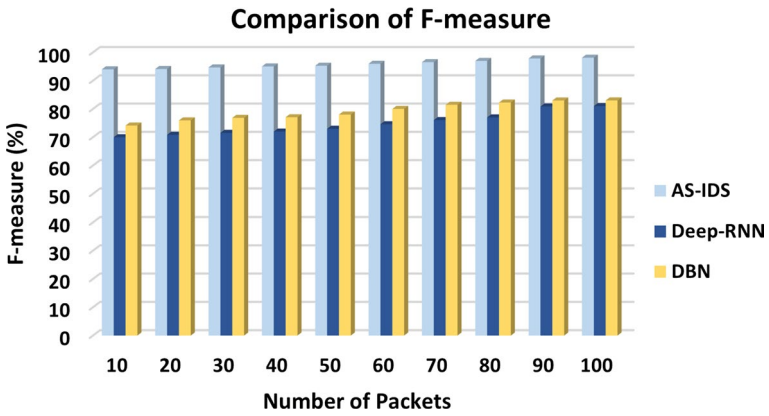

Fig. 8 Score comparison

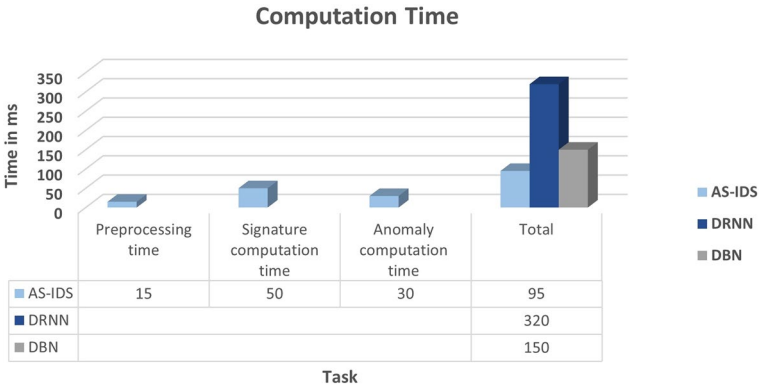

Fig. 9 Computation time

- Our model initially filters the traffic to handle high volume packet streams of IoT traffic, which reduces the overhead introduced by the AS-IDS model.
- Our work stores signatures in the form of a tree using the GST algorithm, which improves intruder detection in the signature-based IDS model.
- HMS-based LightNet is used to detect intruders in the signature-based system, and clustered the incoming packets to reduce the high dimensional feature set. This reduces the time for intruder detection in the signature-based system.
- We considered environment related parameters (SNR and bandwidth) with the anomaly-based IDS model, using the Deep Q-Learning algorithm to differentiate between the attack types.
- The signatures of newly detected attacks in the anomaly-based IDS model updated using the PADS algorithm.

7 Conclusion

In this paper, we proposed a model that combines signature and anomaly-based IDS. The three phases considered here are traffic filtering, preprocessing and hybrid IDS. In the traffic filtering phase, the features of the arrived packet streams are extracted and validated by the IoT gateway,

In preprocessing, the features are converted into numeric values, then normalized and the redundancy is reduced. Preprocessing concentrates the network traffic with the dataset. The traffic packets then enter the hybrid IDS phase, where the signature-based IDS is applied using signature matching and the LightNet algorithm. All unknown packets are processed by the anomaly-based IDS, and the deep Q-learning algorithm considers SNR and bandwidth for attack classification. After results analysis, the proposed AS-IDS model shows greater improvement than other IDS methods. In the future, this AS-IDS system should be extended to the address the following:

- Include additional critical attacks in other datasets and evaluate the performances of the network using deep learning algorithms with optimization.
- Instruction Prevention System (IPS) maybe integrated with our model that will be responsible to take actions against the attacks autonomously using some learning algorithms.
- Provide security to ensure forwarded IoT traffic is from a registered or unregistered user, and apply individual security validation through bio-metric and other authentication methods.

References

1. Jararweh, Y., Otoum, S., Ridhawi, I.Al: "Trustworthy and sustainable smart city services at the edge". Sustain. Cities Soc. **62**, 1–11 (2020)
2. Aloqaily, M., Otoum, S., Ridhawi, I.Al, Jararweh, Y.: An intrusion detection system for connected vehicles in smart cities. Ad Hoc Netw. **90**, 1–14 (2019)
3. Santos, L. Rabadao, C., Gonçalves, R.: Intrusion detection systems in Internet of Things: A literature review. 13th Iberian Conference on Information systems and Technologies (CISTI) (2018)
4. Fu, Y., Yan, Z., Cao, J., Koné, O., Cao, X.: An automata based intrusion detection method for Internet of Things. Mobile Inf. Syst. https://doi.org/10.1155/2017/1750637 (2017)
5. Elrawy, M.F., Awad, A.I., Hamed, H.F.A.: Intrusion detection systems for IoT-based smart environments: a survey. J. Cloud Comput. **7**, 1–20 (2018)
6. Salunkhe, U.R., Mali, S.N.: Security enrichment in intrusion detection system using classifier ensemble. J. Electr. Comput. Eng. (2017). https://doi.org/10.1155/2017/1794849
7. Vengatesan, K., Kumar, A., Naik, R., Verma, D.K.: Anomaly based novel intrusion detection system for network traffic reduction. In: 2nd International Conference on I-SMAC. IoT in Social, Mobile, Analytics and Cloud (2019)
8. Cepheli, Ö., Büyükçorak, S., Kurt, G.K.: Hybrid intrusion detection system for DDoS attacks. J. Electr. Comput. Eng. https://doi.org/10.1155/2016/1075648 (2016)
9. Saleh, A.I., Talaat, F.M., Labib, L.M.: A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. Artif. Intell. Rev. **51**, 403–443 (2019)
10. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine. Electronics **9**, 173 (2020)
11. Khan, I.A., Pi, D., Khan, Z.U., Hussain, Y., Nawaz, A.: HML-IDS: a hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. IEEE Access **7**, 89507–89521 (2019)
12. Elhefnawy, R., Abounaser, H., Badr, A.: A hybrid nested genetic-fuzzy algorithm framework for intrusion detection and attacks. IEEE Access **8**, 98218–98233 (2020)
13. Jiang, K., Wang, W., Wang, A., Wu, H.: Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access **8**, 32464–32476 (2020)
14. Kim, J., Kim, J., Kim, H., Shim, M.: CNN-based network intrusion detection against denial-of-service attacks. Electronics **9**(6), 916 (2020)
15. Tobi, A.M.Al, Duncan, I.: Improving intrusion detection model prediction by threshold adaptation. Information **10**, 159 (2019)
16. Magán-Carrión, R., Urda, D., Díaz-Cano, I., Dorronsoro, B.: Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. Appl. Sci. **10**, 1775 (2020)
17. de Lima Filho, F.S., Silveira, F.A.F., de Medeiros Brito Jr, A., Vargas-Solar, G., Silveira, L. F.: Smart detection: an online approach for dos/ ddos attack detection using machine learning. Secur. Commun. Netw. https://doi.org/10.1155/2019/1574749 (2019)
18. Yang, K., Ren, J., Zhu, Y., Zhang, W.: Active learning for wireless IoT intrusion detection. IEEE Wirel. Commun. **25**(6), 19–25 (2018)

19. Otoum, Y., Nayak, A.: "On securing IoT from Deep Learning perspective", 2020 IEEE Symposium on Computers and Communications (ISCC), pp. 1-7, (2020). https://doi.org/10.1109/ISCC50000.2020.9219671
20. Otoum, S., Kantarci, B., Mouftah, H.T.: On the feasibility of deep learning in sensor network intrusion detection. IEEE Netw. Lett. **1**(2), 68–71 (2019)
21. Khan, Z.A., Abbasi, U.: Reputation management using honeypots for intrusion detection in the internet of things. Electronics **9**(3), 1–30 (2020)
22. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., Alazab, A.: A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. Electronics **8**(11), 1–18 (2019)
23. Iman, A.N., Ahmad, T.: Improving Intrusion Detection System by Estimating Parameters of Random Forest in Boruta. Presented at the (2020)
24. Rajagopal, S., Kundapur, P., Hareesha, K.: A stacking ensemble for network intrusion detection using heterogeneous datasets. Secur. Commu. Netw. (2020). https://doi.org/10.1155/2020/4586875
25. Aung, Y., Min, M.: Hybrid Intrusion Detection System using K-means and K-Nearest Neighbors Algorithms. IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) (2018)
26. Lv, L., Wang, W., Zhang, Z., Liu, X.: A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine. Knowl. Based Syst. **195**, 102548 (2020)
27. Alazzam, H., Sharieh, A., Sabri, K.E.: A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst. Appl. **148**, 1–14 (2020)
28. Mazini, M., Shirazi, B., Mahdavi, I.: Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. J. King Saud Univ. **31**, 541–553 (2018)
29. Zhang, Y., Li, P., Wang, X.: Intrusion detection for IoT based on improved genetic algorithm and deep belief network. IEEE Access **7**, 31711–31722 (2019)
30. Hachmi, F., Boujenfa, K., Limam, M.: Enhancing the accuracy of intrusion detection systems by reducing the rates of false positives and false negatives through multi-objective optimization. J. Netw. Syst. Manag. **27**, 93–120 (2019)
31. Karatas, G., Demir, O., Sahingoz, O.K.: Deep learning in intrusion detection system. International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT) (2018)
32. Thamilarasu, G., Chawla, S.: Towards deep-learning-driven intrusion detection for the internet of things. Sensors **19**(9), 1–19 (2019)
33. Balakrishnan, N., Rajendran, A., Pelusi, D., Ponnusamy, V.: Deep belief network enhanced intrusion detection system to prevent security breach in the internet of things. Internet Things **4**(33), 1–8 (2019)
34. Khan, M.A., Karim, M.R., Kim, Y.: A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. Symmetry **11**(4), 583 (2019)
35. Otoum, Y., Liu, D., Nayak, A.: DL-IDS: a deep learning-based intrusion detection framework for securing IoT. Emerg. Telecommun. Technol, Trans (2019). https://doi.org/10.1002/ett.3803
36. Pajouh, H.H., Javidan, R., Khayami, R., Dehghantanha, A., Choo, K.R.: A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks. IEEE Trans. Emerg. Top. Comput. **7**, 314–323 (2019)
37. Kaur, S., Singh, M.J.: Hybrid intrusion detection and signature generation using deep recurrent neural networks. Neural Comput. Appl. **32**, 7859–7877 (2019)
38. Ye, Q., Zhi, W.: Discrete hessian eigenmaps method for dimensionality reduction. J. Comput. Appl. Math. **278**, 197–212 (2015)
39. Tang, Y., Chen, S.: An automated signature-based approach against polymorphic internet worms. IEEE Trans. Parallel Distrib. Syst. **18**(7), 879–892 (2007)
40. Khan, A.H.: Lightweight Neural Networks. arXiv:1712.05695v1(2017)
41. Mousavirad, S.J., Ebrahimpour-Komleh, H.: Human mental search: a new population-based metaheuristic optimization algorithm. Appl. Intell. **47**, 850–887 (2017). https://doi.org/10.1007/s10489-017-0903-6
42. Yin, C., Zhu, Y., Fei, J., He, X.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access **5**, 21954–21961 (2017). https://doi.org/10.1109/ACCESS.2017.2762418

Yazan Otoum received his M.Sc. in Network Engineering and Management from DePaul University in 2009. Currently, he is pursuing a Ph.D. in Electrical and Computer Engineering at the University of Ottawa. His research interest includes IoT Security, Machine Learning, and Intrusion Detection.

Amiya Nayak received his B.Math degree in Computer Science and Combinatorics and Optimization from University of Waterloo, Canada, in 1981, and Ph.D. in Systems and Computer Engineering from Carleton University, Canada, in 1991. He is now a professor at University of Ottawa. His research interests include mobile computing and Internet of Things.

Authors and Affiliations

Yazan Otoum[1] · **Amiya Nayak**[1]

Amiya Nayak
nayak@uottawa.ca

[1] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada