Discover Artificial Intelligence

Research

Self-healing hybrid intrusion detection system: an ensemble machine learning approach

Sauharda Kushal[1] · Bharanidharan Shanmugam[1] · Jawahar Sundaram[2] · Suresh Thennadil[1]

Received: 7 July 2023 / Accepted: 21 March 2024
Published online: 16 April 2024

Abstract

The increasing complexity and adversity of cyber-attacks have prompted discussions in the cyber scenario for a prognosticate approach, rather than a reactionary one. In this paper, a signature-based intrusion detection system has been built based on C5 classifiers, to classify packets into normal and attack categories. Next, an anomaly-based intrusion detection was built based on the LSTM (Long-Short Term Memory) algorithm to detect anomalies. These anomalies are then fed into the signature generator to extract attributes. These attributes get uploaded into the C5 training set, aiding the ensemble model in continual learning with expanding signatures of unknown attacks. By generating signatures of unknown attacks, the self-healing attribute of the ensemble model contributes to the early detection of attacks. For the C5 classifier, the proposed model is evaluated on the UNSW-NB15 dataset, while for the LSTM model, it is evaluated on the ADFA-LD dataset. Compared to conventional models, the experimental results show better detection rates for both known and unknown attacks. The C5 classifier achieved a True Positive Rate of 97% while maintaining a false positive rate of 8%. Also, the LSTM model achieved a detection rate of 90% while retaining a 17% False Alarm Rate. As the proposed model learns, its performance in real network traffic also improves and it also eliminates human intervention when updating training data.

Keywords Anomaly detection system · Convolutional neural network · Deep learning · Long short-term memory · Recurrent neural network · Self-healing

1 Introduction

With the rapid development and expansion of the Internet and IoT or the Internet of Things, the demand for security against cyber-attacks has also had an exponential spike in recent years. With the growth of the network, the growth of information flowing through is imminent which poses a multitude of opportunities for attackers to act, with malicious intentions [1]. An intrusion can be described as an action that violates the CIA (Confidentiality, Integrity, and Availability) triad of a system by bypassing security put in place with malicious intent. These intrusions can be known or unknown, the latter has been coined as zero-day attacks [2]. A constant monitoring system that dynamically scans

Sauharda Kushal, Jawahar Sundaram and Suresh Thennadil have contributed equally to this work.

✉ Bharanidharan Shanmugam, Bharanidharan.Shanmugam@cdu.edu.au; Sauharda Kushal, sauharda92@gmail.com; Jawahar Sundaram, sundaramj@caias.in; Suresh Thennadil, Suresh.Thennadil@cdu.edu.au | [1]Energy and Resources Institute, Faculty of Science and Technology, Charles Darwin University, Darwin, NT 0815, Australia. [2] Computer Science and Applications, Christ Academy Institute for Advanced Studies, Bengaluru, India.

events within a system and looks for inconsistencies, preventing any such events from entering the system is generally referred to as an Intrusion Detection System (IDS) [3].

There are two types of intrusion detection: Signature based detection and anomaly-based detection. The signature-based network intrusion detection system uses a signature-driven database of attack signatures, which are matched to real-time network traffic for the detection of intrusions. This database is usually updated for better results. Therefore, Signature-based Network Intrusion Detection Systems (SNIDS) are also referred to as real-time intrusion detection systems. However, an anomaly-based intrusion detection process relies on classifiers that separate benign network and system behaviours/activities from unknown or unusual activities/behaviours by studying the normal behaviours of the system and network traffic [4]. SNIDS analyse all network traffic and detects attacks such as Denial of Service (DoS) and many more, whereas Anomaly-based Intrusion Detection Systems (AIDS), work independently on host devices or workstations monitoring packet contents and system log files to identify any abnormality and inconsistency in system activities [5]. The anomaly-based IDS (AIDS) identifies anomalies by analyzing the normal network behavior. Anomalies can be detected by observing the deviations between real-time network activities and a normal behaviour pattern [6].

An attack of the same origin can be identified by its signature and used as a reference for similar attacks. SIDS generally displays high accuracy but fails to detect zero-day attacks. As a result of malware's polymorphic behavior, SIDS relies on signature databases which can be circumvented. The malware evolves over time, making it difficult to match signatures in the database [7]. Signatures are strings, patterns, or rules that show similarities to known attacks. A SIDS compares the signature with real-time network traffic to identify any intrusions in the network based on similarities with attack signatures. A database of pre-existing signatures can mitigate known attacks [8]. It is also difficult to maintain a low false positive rate for anomaly-based IDS. A lack of behavioral studies and improper algorithms and processing schemes in IDSs contribute to this challenge [9]. Anomaly-based IDS helps detect unknown attacks but are highly unreliable with the results they produce when compared with a signature-based IDS.

Hybrid intrusion detection methods have proven to resolve the issue of the two conventional, individual intrusion detection methods. However, there are a few areas that pose opportunities for improvement in the Hybrid Intrusion Detection framework. The proposed model incorporates machine learning algorithms and deep learning algorithms that are most suitable for intrusion detection models. The performance superiority of the C5 classifier was demonstrated by the researchers which yielded better results than K-Nearest Neighbour (KNN), Random Forest, Naïve Bayes, and Classification And Regression Tree (CART) [10]. Also, Imrana et al. [11] proposed IDS with conventional LSTM trained on KDD 99 which yielded better results than J48, Naïve Bayes, Random Forest, (recurrent neural network) RNN, SVM (Support Vector Machine) and Standard Template Library (STL). Furthermore, hybrid intrusion detection techniques incorporate anomaly and misuse detection approaches to detect both known attacks and unknown attacks, as well as generate signatures for zero-day attacks or attacks that aren't in the signature repository and update the IDS repository [12]. Zero-day attacks can be used to overcome misuse detection shortcomings. Additionally, detecting attacks in the first phase reduces anomaly detection system stress.

Finally, datasets are key to Machine Learning (ML), in its training and testing phases. Updating algorithms and models and including current attack features will ensure the efficacy and reliability of IDSs. Dataset training and testing will help the ML models identify new attacks, and a large dataset will confirm patterns, sequences, and rules in network behavior [13]. KDD 99 and NSL-KDD 99 datasets are used in most works on IDS and Hybrid Intrusion Detection Systems (HIDS). However, these datasets are outdated as they do not include new attacks and require heavy pre-processing. The proposed model is trained with UNSW-NB15 and ADFA-LD, which embodies new attack types. The clustering technique supported by UNSW-NB15 can also be used to analyze the similarity of attributes and adjust attribute usage to improve the performance of IDSs [14].

Machine Learning (ML), in intrusion detection and prevention, has proven pivotal as witnessed by its prevalence in cyber-attack detection. The ML models are tasked with identifying data patterns or predicting behavior, by processing vast-scale datasets collected over an extensive period [15]. Most intrusion detection systems prevalent today use ML-based algorithms as their detection strategy [16]. Machine Learning can generally be categorized into shallow learning and deep learning. Deep learning models, in intrusion detection systems, are predominantly neural network models with a high number of hidden layers. Such models are capable of learning highly complex non-linear functions and the hierarchical layer structure facilitates the learning of useful features from the data. Deep learning has established a firm grip on Intrusion Detection Methods [16].

A self-healing component [17] is incorporated into the proposed model, allowing it to learn signatures from anomalous packets by using techniques that combine signature-based and anomaly-based IDS models, as well as a self-learning attribute for the ensemble model to learn. UNSW-NB15 and ADFA-LD are the two models used in the current IDS framework to train the models, which are very relevant for the current IDS framework.

Aims of this research and contributions

1. The following are the list of objectives in this research.
2. The research aims to establish a network intrusion detection system that incorporates some of the best machine learning models in terms of performance.
3. The proposed model also aims to develop a hierarchical structure of cyclic dataflow that enables the system to become a self-sufficient IDS.
4. The proposed hybrid intrusion detection system aims to achieve a high detection rate and high accuracy by incorporating a self-learning technique through a signature generator.
5. Anomalies identified from the anomaly-based IDS are fed into a signature extraction based on their deviation from normal traffic patterns and similarity to malicious traffic patterns.
6. The self-learning framework significantly increases the detection rate of SIDS in the hybrid architecture by updating the signature repository with unknown anomalous signatures.
7. The proposed model aims to grow the signature repository over time by collecting attributes and signatures of unknown attacks.
8. The model encompasses continual learning without the need for human intervention to update the signature repository.

The main contributions of this paper are as follows:

1. In this paper, we propose a hybrid intrusion detection system with self-healing attributes, machine-learning models, and a robust architecture.
2. The model has been trained and tested using relevant datasets pertaining to today's attacks using relevant datasets.
3. In order to detect intrusions, the system proposes a continuous learning hybrid intrusion detection model, in which attack signatures are continuously updated without the need for human intervention.
4. During real-world network deployments, the self-healing model improves network performance through the continuous extraction of signatures from anomalies detected in the network over time.
5. The proposed system contributes to a continuous learning system that eliminates the need to update training datasets for updated attack tracking in the foreseeable future since the architecture of data flow within the proposed model facilitates continuous learning.

The rest of the paper is organised as follows: Sect. 2 consists of a summary of the closely related works, and Sect. 3 consists of a background study based on selected systems. We have explained the proposed methodology in Sect. 4, as well as the results and benchmarking in Sect. 5. In Sect. 6, we have explained the conclusions and suggested the next steps.

2 Related works

Studies in IDS demonstrate the variations in the categorization of NIDS based on patterns, rules, statistics, states, and heuristics. Also, there have been numerous studies that have proposed Hybrid NIDS with different techniques, models, and architectures combining signature-based and anomaly-based detection systems. Moreover, various Shallow Machine Learning models, Deep learning models, and hybrid models (in terms of the algorithm used rather than network and system parameters) have also been used in conjunction with Hybrid NIDS [18]. SNORT's performance has been highlighted which uses an anomaly pre-processor integrated with the SNORT IDS which compensated for the shortcoming of SNORT IDS, on itself, as it was incapable of detecting unknown or zero-day attacks [19].

The application of Neural Networks extends widely into different sectors, especially in agriculture. In the paper [20] in which MaskRCNN, a fast convolutional neural network model is employed in detecting weed which was trained using a custom leaf-based dataset. The experiment incorporated 50 datasets, more than 100 training configurations, and 300 h of training of the datasets on the MaskRCNN model. The deep learning model yielded a 93% of mAP accuracy in the detection of training and 95% accuracy in the testing images. The use of the detection model will save farmers time and effort in detecting weeds in a pasture environment. Similarly, the use of the deep learning model in anomaly detection is not limited to Intrusion Detection Systems but has several many industries dependent on it. One case of Machine Learning models used in detecting anomalies is Energy consumption. The research uses deep learning models to detect anomalous behaviour of energy consumption to detect abnormal energy consumption behaviour and help prevent it. Using various features such as temperature, humidity, occupancy, and so on, clusters were generated. The features considered as inputs represent the context of the energy meter. The meters are then grouped by the context type and the behaviour gets analysed. Through the formation of a cluster using the KNN algorithm, any deviation from the cluster is identified as an anomaly as those instances represent abnormal behaviour. Such deviations are then identified and investigated further [21].

One of the biggest challenges in the health sector is the classification of imbalanced miRNA (micro-Ribonucleic Acid) sequences. miRNA helps detect and diagnose cancer. Jain et al. [22] proposed a Hybrid Neural Network model with Deep ANN and Deep Decision Tree classifier. The proposed hybrid method performed better than the individual Neural Network and the Decision Tree model. The proposed model showed an accuracy of more than 99% and improved the time complexity when compared with other existing models [22]. There have been several innovations to overcome the security issues affecting data integrity and privacy, one of which is through the use of blockchain technology. The paper explained the use of blockchain in various sectors that contributes to maintaining the reliability of data. The paper illustrates the benefits of integrating blockchain technology with IoT through decentralization, which is considered one of the most significant factors as a single authority cannot authorise a transaction but require a bulk of participants to do so. [23] The application of blockchain in an IDS can result in the elimination of opportunities to tamper with the alerts generated by the IDS. This can be achieved by using blockchain technology where all the alerts generated by the IDS are treated as a transaction. The collective alerts then adopt a consensus rule which helps validate the alerts before being placed in the block [24]. The use of private Blockchain models in IDS allows the network owners to privately invite and vet the participating nodes.

Shallow learning, more specifically, classifiers, has had a huge impact on network intrusion detection. The proposed Hybrid IDS by Tesfahun and Bhaskari [25] was a layered approach to HIDS consisting of 2 layers: one being a misuse detector or a signature-based NID model and the other layer functioning as an anomaly-based NID. The SNIDS was based on a random forest classifier model and the AIDS was built using the bagging technique with an ensemble of one-class support vector model classifiers and the dataset used for the study was NSL-KDD. The proposed model produced an attack detection rate of 92.1% and a false positive rate of 6.4%. Furthermore, Chitrakar and Chuanhe [26] developed a novel ensemble HIDS that comprised a combination of the C5 classifier and OCSVM (One Class Support Vector Machine) classifier which hosted both signature-based NIDS and anomaly-based IDS. In contrast with results from SIDS and AIDS, the proposed HIDS gave a higher detection rate and a lower number of false positives. The datasets used to evaluate the proposed HIDS were NSL-KDD and ADFA. The proposed technique yielded the highest NSL-KDD accuracy when compared with other techniques such as C4.5, Random Forest, KNN, and Naïve Bayes [26]. To overcome the drawbacks of an individual learning algorithm, multiple machine learning algorithms are used, to complement the overall Intrusion Detection process. The accuracy of the developed model was 83.24%.

The prevalence of deep learning in recent years has been attributed to various reasons each unique in its scope. Firstly, the processing capabilities have drastically improved due to powerful GPUs, also known as Graphics Processing Units, and the ease of acquiring the services, and the cost of GPU providers [27]. Secondly, the cost of hardware dropping significantly in the past decade has paved a path for increasing deep learning Approaches. deep learning algorithms' ability to form learnable links between actions and effects, also known as Depth of Credit assignment paths, and what differentiates deep learning models from Shallow Learning models has led to its increasing usage [28]. With regards to deep learning models being used in a recent IDS, Khan et al. [29] also proposed a HIDS based on a Convolutional-LSTM network model which was also a two-stage IDS in which the first stage employed an anomaly-based Intrusion Detection model that was based on Spark ML whereas the second stage was a misuse detection model based on the Conv-LSTM network. The dataset used was ISCX-UNB. An accuracy of 97.29% was observed in detecting network misuse under the proposed HIDS.

The architecture of the NIDS also plays a vital role in the performance of the hybrid intrusion detection model [30]. As [31] developed a novel hybrid detection method that integrated the misuse detection model and anomaly detection model hierarchically in a decomposition structure in which the SNIDS was built based on the C4.5 decision tree algorithm whereas the AIDS was built based on multiple one-class SVM models created for the decomposed subsets created by the SNIDS. The models were evaluated through experiments on the NSL-KDD dataset. The proposed decomposition structured model yielded a high detection rate and low false positives in comparison to previous studies. However, it also displayed incredibly low training and testing time when compared with the serial conventional hybrid model and the parallel conventional hybrid model. The proposed hierarchical model produced an accuracy of 99% and a false positive of 2%. More advancements and variations in HIDS have been developed in recent years. This is along with the proposal by Kim et al. [32] that conceptualized a signature generation engine integrated with a deep recurrent neural networks based HIDS. The proposed HIDS comprised a signature detection system, a Deep Neural Network-based anomaly detection system, and a Signature Generation Engine (SGE) which was envisioned to sustain the Detection approach as the generated signatures were fed into the signature repository. The results were: an updated & extensive signature repository, simultaneous detection and signature generation of unknown attacks, and a self-healing intrusion detection approach [32].

In some studies, multiple software has been integrated into one another to form a hybrid intrusion model. This model fundamentally works similarly to other HIDS, a combination of AIDS and SIDS. The proposed HIDS by Rizvi et al. [33] comprised of a combination of Packet Header Anomaly Detector (PHAD) and Network Traffic Anomaly Detector (NETAD) integrated into signature-based IDS Snort. PHAD uses a host protocol model and time-based model, while NETAD uses a host packet model. As a result, the HIDS was able to detect an additional 119 attacks that the traditional Signature-based detection of SNORT could not [34]. More recently, Degeler et al. [35] proposed a hierarchical hybrid intrusion detection approach with an anomaly detector as the first stage of the IDS and an attack classifier as the second stage of the IDS. The anomaly detection is done via a novel lightweight solution based on Multi-modal Deep Autoencoder (M2-DAE) and the attack classification is carried out via soft output classifiers. This approach follows an inverted hierarchical architecture in contrast with the predominant studies in IDSs. The M2-DAE as a result displayed a decline in false positive rate by 40% in comparison with multiple baselines at the same positive rate. Additionally, the HIDS in comparison with best-performing misuse detectors showed an increase in the F1 score by 5% [36]. Similarly, to improve the efficiency and accuracy of a Hybrid Intrusion Detection System, Sohi et al. [37] proposed an IDS: Hybrid VMM-based Honeypots integrated into the HIDS that transform the entire IDS into a self-healing Intrusion Detection Prevention System (IDPS). A unique component of the proposal is the IDPS signature and anomaly databases, as well as the Intrusion Detection Prevention Operations Centre (IDPOC), which allows users to quarantine potential threats or ban traffic from a particular source [33].

The paper proposed by Creech and Hu [38] introduces a self-healing intrusion detection system with a danger theory that investigates the danger signals which the IDS perceives as malicious, firstly in a manual observation by the system's operator and secondly in automated observation done by analysing system logs such as events considered intrusive, sudden spike in CPU usage, packet loss, undefined usage of ports and so on. In both cases, if the events are confirmed to be dangerous, they are communicated to the entire network, so that every device can check its timeline for similar events. The authors concluded that such self-healing IDS could result in a drastic decline in false positives [35]. In addition, recent studies have proposed IDSs that generate synthetic signatures that can be used to detect zero-day attacks, such as the work of [39] that uses RNN, also known as recurrent neural network, to develop synthetic signatures and mutants of known attacks. Through the development of a mutation signature database as well as synthetic signatures through deep learning, the IDS proposed in the study defends against known and unknown attacks [37].

Several studies suggest that using the KDD99 dataset in anomaly detection techniques is insufficient to capture the wide spectrum of attacks that exist today. Chew et al. [40] conducted a comparative study highlighting the complexity of the ADFA feature in contrast to the relatively simple feature of the KDD99. Moreover, the training algorithms with older datasets not only could not detect contemporary attacks but also are not very reliable. This is because they are not as rich in data as the newer ones [38]. UNSW-NB15 is synthetic data, like CIC_IDS2017 whereas ISP and UQ are real-world data. The binary classification of attacks was highly accurate when running the UNSW-NB15 dataset [39]. Also, a study conducted by Vinayakumar et al. [41] demonstrated that CIDDS001 suffers from high false positives through 10 machine models classification whereas both UNSWNB15 and GureKDDCup obtained low false positives and high accuracy rate [40]. The comparision of various network intrusion datasets is represented in Table 1 for different types of attacks.

The HIDS have been trained and tested using UNSWNB15 and ADFALD as datasets such as KDD99 and NSLKDD do not reflect relevant results in regard to accuracy and detection rate as they lack modern attack patterns with low congestion and also lack the normal traffic behaviour of the present [42]. The HIDS proposed by some authors combines the anomaly and signaturebased IDS to develop a HIDS without learning the signatures of the anomalies. The proposed HIDS is a self-learning IDS that extracts signatures from the anomalies found and transfers them to the SIDS, allowing the early detection of previously unknown threats [15]. The IDS proposed hard-lined the performance superiority of the deep learning approach, which demonstrated dominance in terms of accuracy, precision, recall, and F-score of datasets: KDDCup99, NSLKDD, UNSWNB15, WSNDS and CICIDS 2017. This study evaluated the performances of shallow and deep learning models with a cross matrix of their performance with all the 5 datasets [41].

Table 1 Comparison of network intrusion datasets

Datasets	Year	Dataset feature count	Attack types
KDD99	1998	43	DoS, R2L, U2R and probilng
NSL-KDD	1998	43	Normal, DoS, R2L, U2R and probing
DARPA	1998	43	DoS, R2L, U2R and probilng
KYOTO	2006–2009	23	Oth, rej, rsto, rstos0, rstr, rstrh, s0, s1, s2, s3, sf, sh, shr
ISCX	2012	80	Attacker and normal
AWID	2015	156	Normal, Flooding, injection and impersonation
UNSW-B15	2015	49	Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms
CSE-CIC	2018	80	DoS Golden Eye, Benign, DoS hulk, DoS Slow http, DoS Slowloris, DDoS-LOIC HTTP, DDoS-LOIC-UDP, DDoS-HOIC, SSH-Patator, FTP Patator, Brute force, XSS, Botnet, infiltration, SQL injection

Furthermore, LSTMRNN yielded better results when compared to feed forward neural networks (FNN), generative neural networks (GNN), and recurrent neural networks (RNN) with hessianfree and Jordan ANN (Artificial Neural Network) in the experiment done by Kim and Kim [43]. LSTMRNN in comparison with conventional RNN resolves the issue of vanishing gradient which conventional RNN suffers from. Also, LSTM-RNN learns long-term dependencies by using a gating mechanism. It also holds previous states in its memory cell [44]. The selection of LSTMRNN for AIDS in the proposed HIDS was done because of the features mentioned in [44] paper. The proposed HIDS has been heavily influenced by the IDS model in which the SIDS matches the signatures in Lightnet utilizing an HMS or Hybrid Multi-Start algorithm and the AIDS utilizing Deep Q-learning. The proposed framework also is self-healing as the signature from the anomalies is fed into the signature repository [45]. The selection of the C5 classifier was done with regards to the proposed performance evaluation by Tang et al. [46] which showed 99% accuracy and detection rate with a shallow learning model as a SIDS. Belavagi and Muniyal [47] This cemented the fact that the use of deep learning is not always required for yielding better results as similar results can be obtained from simple MLs. While Intrusion Detection approaches possess a variety of concepts and models, there is still much-unexplored territory that needs study and innovation due to the ever-growing interconnectedness of the Internet. Table 2 illustrates comparison of related works for detection category, machine learning algorithm, datasets with paper title and Table 3 gives information about comparison of proposed model and related works performance.

3 Background study

This section of the study discusses the models which have been selected for the HIDS and the datasets which are being used to train and test the model. It also involves the mechanism of the selected models with a comparison to some other algorithms, explaining the advantages of the selected models over others. Lastly, it involves the details of the datasets: UNSW-NB15 and ADFA-LD.

3.1 Long short-term memory-recurrent neural networks

A sequential neural network works by processing inputs independently from each other, however, in case of RNNs, inputs are considered in context and interdependence between inputs is reflected. RNN, a deep learning algorithm that incorporates inputs, outputs, and hidden layers, allows the entire network to be stored and remembered. RNN has a one-directional flow in a loop that can memorize the previous information, then apply the rules to the current output. This differentiates RNN from Feed-Forward Neural Networks. The nodes between the hidden layer also have connections and the previous output is related to the current output. Moreover, the output of the hidden layer acts as the input of hidden layers [5]. RNN, in an IDS, identifies patterns and irregularities in a huge dataset that helps establish rules for evaluating real-time network traffic for malicious or normal activity [46].

A recurrent neural network in Fig 1 used for attack classification employs sequential layers which perform information processing in feature representation. This was only made available in recent years due to the affordable hardware and

Table 2 Comparison of related works

References	Detection category	Machine learning algorithm	Datasets	Paper title
[21]	Anomaly, SNORT	N/A	N/A	Design of a Snort-Based Hybrid Intrusion Detection System
[27]	Anomaly, signature	Random Forest classifier, one class support vector	NSL-KDD	Effective Hybrid Intrusion Detection System: A Layered Approach
[3]	Signature and anomaly	C5 decision tree, one class support vector	Bot-IoT	A Novel Ensemble of Hybrid IDS for Detecting IoT Attacks
[31]	Anomaly, misuse	Spark ML, Conv-LSTM	ISCX-UNB	A Scalable and Hybrid IDS Based on the Convolutional-LSTM Network
[36]	Misuse, anomaly	C 4.5 decision tree, one class SVM	NSL-KDD	A novel hybrid intrusion detection method integrating anomaly detection with misuse detection
[13]	Anomaly, signature generation	LSTM	DARPA 1998, ISCX 2012, NSL-KDD	Hybrid intrusion detection and signature generation using Deep Recurrent Neural Networks
[37]	Misuse, anomaly	SNORT	IDEVAL	A hybrid intrusion detection system design for computer network security
[39]	Anomaly, signature	M2-DAE	Bot-IoT	A Hierarchical Hybrid Intrusion Detection Approach in IoT Scenarios
[35]	Signature, anomaly	SNORT	N/A	Advocating for Hybrid Intrusion Detection Prevention System

Table 3 Performance based comparison of proposed model and related works

References	Signature extraction	Detection rate	False alarm rate
[3]	No	N/A	N/A
[25]	No	92.10%	6.40%
[38]	No	SIDS: 97.2% (ADFA) AIDS: 65.7% (ADFA)	SIDS: 2.5% ADIS: 10%
[29]	No	97.27%	0.70%
[31]	No	99%	2%
[13]	Yes	N/A	N/A
[33]	No	N/A	N/A
[35]	No	90.99%	1%
Proposed model	Yes	SIDS: 97% AIDS: 90%	SIDS: 8% NIDS: 17%

availability of high processing capabilities for general/research use. The proposed RNN model is a multilayer Long-Short Term Memory that outperforms most traditional approaches in IDS. RNN is a deep learning model widely used in recognizing generated images and text and interpreting the results. However, the failure to capture long-term dependency in RNN can be resolved by LTSMRNN. LTSM shown in Fig 2 is exclusively developed to overcome the problem of long-term dependency. The disappearing gradient issue in RNN can be resolved by achieving disappearing gradient descent, an algorithm for optimization that finds the neural network weights to avoid long-term dependency [48].

3.2 C5 decision tree

The decision tree, in its simplest form, is an if-then-else rule-based machine learning algorithm that is a very powerful classifier and has had a very high detection rate in various sectors of application. The C5 decision tree can deal with missing attributes by providing value to those attributes that are most common in other instances at the same node [49]. In a decision tree, each branch node represents a selection of alternatives, and each leaf node corresponds to a

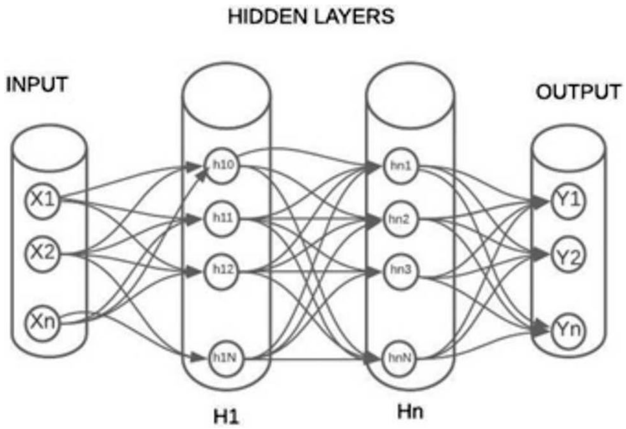


Fig. 1 Recurrent neural network [46]

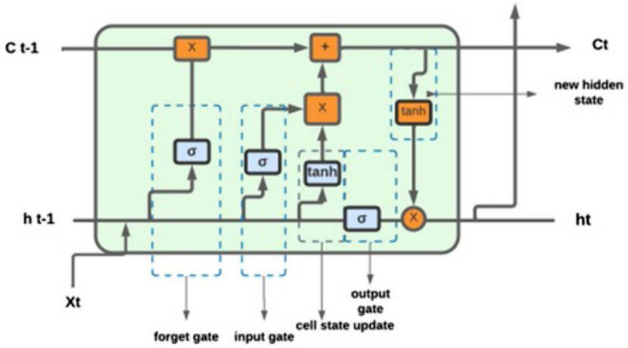


Fig. 2 Long short-term memory [48]

classification/decision [50]. And each decision tree represents a rule. However, C5 supports decision tree boosting which helps in generating and combining multiple classifiers for improved prediction [49].

C5 follows the algorithm of its predecessor, C 4.5, and has features such as the large-scale decision tree which makes it easy to understand with a visual representation of the rules. The missing value while training the algorithm will also be handled within C5. Missing values will be marked as '?' and will not be used in gain and entropy calculations. The C5 classifier resolves the problem of overfitting data in the decision tree by:

1) Stop the production of the decision tree once it reaches the point where the training data has been perfectly classified.
2) Applying Post prune to the tree when there is overfitting of the training data.

In post-pruning, a decision tree is pruned after it has been constructed, such as when a decision tree has very deep levels of branching, in which case post-pruning may be used to speed up the process. The C5 classifiers are used to select a small subset of relevant features from the datasets provided, which have been shown to perform well even with data that has high dimensionality. A high-dimensional problem has been one of the challenging factors in the design of a variety of other machine learning algorithms [51].

3.3 Datasets

Datasets are collections of data that have been gathered and organized in such a way that they are commonly processed and analyzed. The following datasets have been used for testing and benchmarking the proposed model.

i) UNSW-NB15

UNSW-NB15 is a dataset from the University of New South Wales (UNSW) in Australia, which shows network intrusion detection using behavioural analysis. This dataset consists of a hybrid collection of real modern normal activities and synthetic prevalent attack activities which holds nine attack behavior types Fuzzers, analysis, backdoor, DoS, exploits, generic, reconnaissance, Shellcode, and worms. The datasets have been partitioned for training and testing ML [13]. The categorization of UNSW-NB15's features by type is summarized in Table 4 with total number and names.

ii) ADFA-LD

The ADFA consists of AIDS-based data. These datasets cover both Linux and Windows operating systems. The data collection for Linux includes system call traces which when used for the training set, traces of 300 bytes to 6kB were neglected. Similarly for validation or testing set traces of 300 bytes to 10 kb were neglected. For Windows, there are DLL or Dynamic Link Library calls of 1828 normal traces and 5773 attack traces [52]. ADFA-LD has been used in training the LSTM model. The Table 5 shows categorization of ADFA datasets by data types for windows and linux platforms.

4 Methodology

In this section, the proposed architecture and the processes have been defined. The C5 classifier is built upon R studio whereas the LSTM model has been built on Keras through Python. The details of the flow of packets and the decision points have been highlighted in Figs 3, 4 and 5.

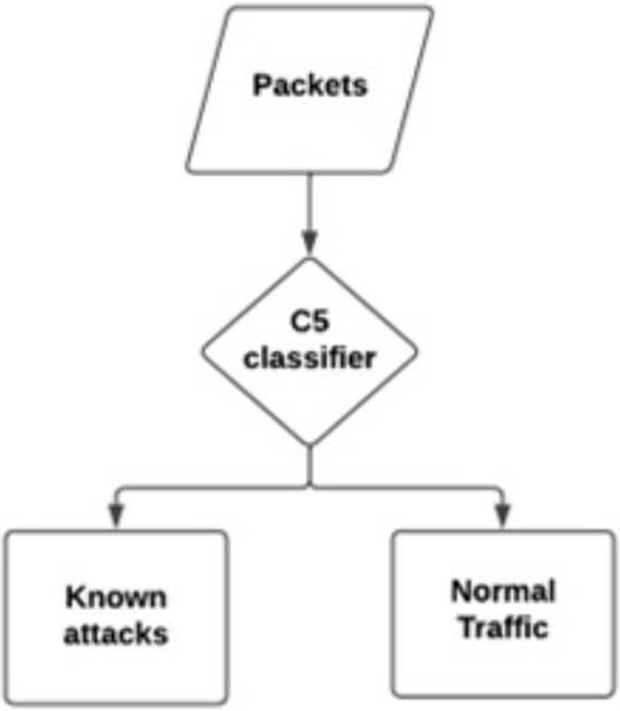
Table 4 categorization of UNSW-NB15's features by type

Feature type	Total number	Names
Binary	3	is_sm_ips_ports,is_ftp_login,Label
Float	10	dur, Sload, Dload, Sjit, Djit, Sintpkt, Dintpkt, tcprtt, syncack, ackdat
Integer	28	Sport, Dsport, Sbytes, Dbytes, Sttl, Dttl, Sloss, Dloss, Spkts, Dpkts, Swin, Dwin, Stcpb, Dtcpb, Smeansz, Dmeansz, trans_depth, res_bdy_len, ct_state_ttl, ct_flw_http_mthd, ct_ftp_cmd, ct_srv_src, ct_srv_dst, ct_dst_itm, ct_src_itm, ct_src_dport_itm, ct_dst_sport_itm, ct_dst_src_itm
Nominal	6	srcip, dstip, proto, state, service, attack cat
Timestamp	2	Stime, Ltime

Table 5 categorization of ADFA datasets by data types

	ADFA-WD (Windows)		ADFA-WD (Linux)	
Dataset	Traces	System calls	Traces	System calls
Training data	355	1,35,04,419	833	3,08,077
Validation data	1827	11,79,18,735	4372	21,22,085
Attack data	5542	7,42,02,804	746	3,17,388
Total	7724	20,56,25,958	5951	27,47,550

Fig. 3 Packet flow



The proposed hybrid IDS in Fig 6 is built on highly effective individual network intrusion detection models. The Signature-based IDS is based on the C5 decision tree algorithm which classifies the inputs into known attacks and normal packets and is one of the highest accuracy-yielding algorithms in network intrusion detection. Similarly, the LSTM-RNN algorithm is used in anomaly-based intrusion detection as it is class-leading, in terms of performance, for determining anomalies from normal activities. The hybrid approach helps in the detection of known as well as unknown attacks. Moreover, the self-healing attribute of the proposed hybrid intrusion detection system assists in storing signatures of anomalies detected by AIDS. This helps in the early detection of similar attacks in the future through signature matching. Due to the increasing volume of attacks through circumvention techniques such as polymorphism that changes the signature of malicious packets, anomaly-based detection has been integrated into the IDS which is able to detect known as well as zero-day attacks. ML algorithms have been chosen after considering previous research which yielded one of the highest performance metrics related to intrusion detection. As a result, a C5 classifier model was implemented for binary classification as a signature-based intrusion detection model, whereas an LSTM model was implemented as an anomaly-based detection model. The anomalous packets are then evaluated, and features are extracted into a '.csv' file. These attributes from the anomalous packets when verified as malicious are categorized as attacks and fed into the decision tree. This helps the proposed HIDS in detecting similar attacks at an earlier stage and has been proposed as a self-healing approach. Similarly, attributes extracted from zero-day attacks, which have been detected externally, can also be fed into the SIDS stage of the proposed IDS.

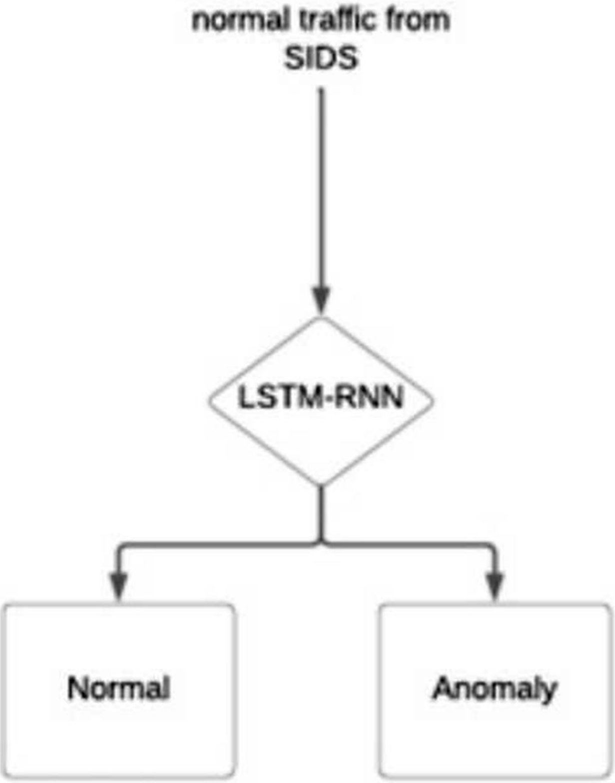


Fig. 4 Flowchart for signature-based IDS

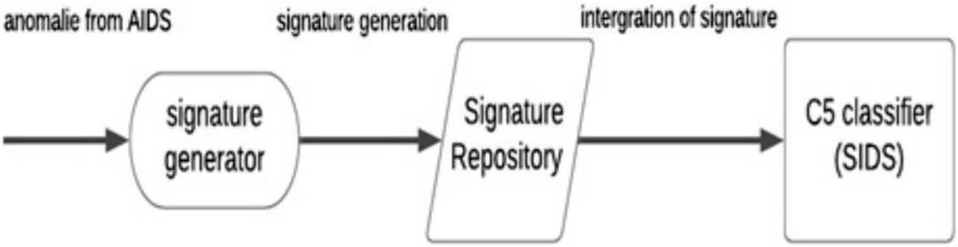


Fig. 5 Flowchart for anomaly-based IDS

4.1 Proposed hybrid intrusion detection system

The proposed hybrid intrusion detection system will combine misuse detection and anomaly detection with a high detection rate and a continuous signature generator. This will add newly produced signatures to the signature repository. The proposed HIDS not only resolves the issues of SIDS and AIDS individually but also creates a flow of newly discovered signatures of unknown attacks into the signature repository. This contributes to the early detection of those attacks. Such a feature helps in early detection, which saves time as well as optimizing resource utilization. The HIDS can be segregated into three stages:

a) Signature-based intrusion detection model utilizing the C5 algorithm (*stage 1*)
b) Anomaly-based intrusion detection model utilizing the LSTM recurrent neural networks algorithm (*stage 2*)
c) Signature generator (*stage 3*)

The 3 staged HIDS have a combination of the C5-based Signature-based Intrusion Detection model and Recurrent Neutral Network-based Anomaly Detection model in a hierarchical order. The signature generator then characterizes and extracts signatures from, the detected anomalies which then get updated in the signature repository used by the SIDS. The subsequent benefit would be the discoverability of the recently detected anomalies in the earliest stage of HIDS.

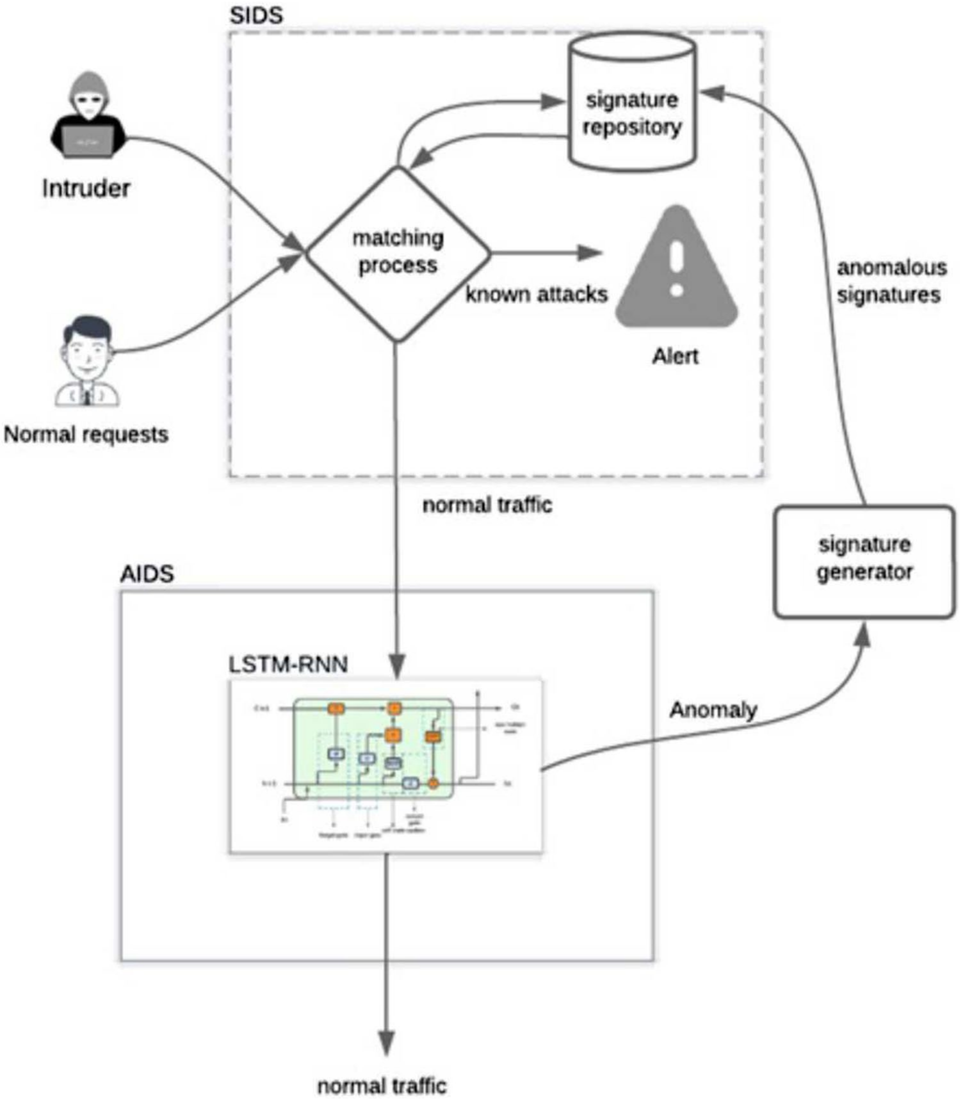


Fig. 6 Proposed HIDS

Stage 1

SIDS has been placed at the upper level in the hierarchy of the HIDS because of the signature repository that the detection system hosts. Also, the early detection of known intrusions, through signature matching reduces the load on the anomaly detection method in the latter stage. Since SIDS provides low false positives and high accuracy in detection, signature detection must be the first stage rather than the latter. This is because signature detection is a progressive narrowing of attacks in the proposed IDS. A reciprocal to the proposed hierarchy could lead to high redundancy of the SIDS (if placed in the second or third stage). It is believed that eliminating the known attacks in the first stage can lead to reduced resource utilization in the AIDS field and possibly save time as well.

Ahmad et al. [7] compared the effectiveness of signature-based anomaly detection with the C5 classifier in comparison with other Machine Learning algorithms and established that there was a reduction in the false negatives and a significant improvement in the rate of detection. The classifiers were trained and tested using NSL_KDD. Ahmad et al. [7] Furthermore, the UNSW-NB15 datasets will be run through the C5 decision tree algorithm along with other ML algorithms in the methodology and the results will be analyzed for comparison of accuracy and false alarm rate. RStudio and WEKA will be used to train and test the C5 model. In this SIDS, unknown packets are handled through signature matching in determining the nature of the packets i.e., normal, or abnormal. When the signature extracted from the packets match with one in the signature repository, an alert will be triggered which will be reviewed by the user. However, if there is no match the packet will be forwarded to AIDS.

Stage 2

The packets categorized as normal, by the SIDS will be the input data in the Anomaly Detection phase. The SIDS is responsible for building a normal behavior profile that represents the pattern and summary statistics of network traffics which are non-malicious. For the training phase, an offline component will be used to help build the profile for normal user behavior, through the extraction of rules from network traffic that are labeled as non-attacks. Similarly, the SIDS will also learn the attack classes, in an offline component, through the network traffics labeled with known attacks [12]. The proposed system uses ADFA-LD and ADFA-WD. datasets used to train the LSTM-RNN.

The training datasets are used to train the classifiers whereas the testing datasets are used to measure the accuracy of the classifier. The classification conducted is binary which produces two classes that are either normal or anomaly. Sarhan et al. [13] implemented an IDS based on LSTM-RNN, which was trained using instances from KDD Cup 1999 dataset. The result demonstrated a superior detection rate and accuracy when compared to the performance of GRNN (General Regression Neural Network), PNN (Probabilistic Neural Network), RBNN (Radial Basis Functions Neural Networks), KNN, SVM, and Bayesian [32]. Also, Naidu and Avadhani et al. [50] proposed an LSTM RNN utilizing an Adam optimizer, that yielded an accuracy rate of 99.97% as a binary classifier IDS in Anomaly detection. Based on the classifiers the packets then get matched against the normal behavior profile and if it detects any deviation in the pattern, an alert gets sent to the user. Those packets then get sent to the signature generator which has been categorized as malicious packets [48].

Stage 3

The signature generator is an integral phase in this proposed hybrid detection method due to the conversion of features extracted from anomalous packets that are segregated by the AIDS into signatures that help identify the attack about the abnormality of the anomaly. This conversion relies on the learning capability of the signature generator primarily based on the features of the anomalous packets. The generated signature is then fed into the signature repository which aids in more effective, precise, and accurate detection of future attacks by the proposed HIDS [53]. AIDS provides the rate of normality and the rate of abnormality of each connection after the processing of anomaly detection. The rate of normality refers to the similarity of features of the packets to the normal traffic behavior whereas the rate of abnormality refers to the degree of deviation of the features of the processed traffic with the normal traffic.

The signature generation proposed by Hwang et al. [54] was a weighted signature generation where the rate of normality (normality score) and rate of abnormality score (anomaly score) were normalized, in this case, the sum of the scores was 1. They defined the overall rate of normality and abnormality of a pattern as the sum of the normality score and anomaly score of all the established connections that match the pattern [54]. Signatures transferred into the repository are those which have a high rate of abnormality and a low rate of normality. A high rate of abnormality of a signature would be the result of more anomalous connections matched. Whereas a low rate of normality of a signature is a result of less normal connections matched. A low rate of normality also results in lesser false alarms. This is due to the high deviation of the signature from the ones with normal traffic. The signatures with a low rate of normality and high rate of abnormality are then integrated into the signature repository in stage 1.

Firstly, the packets are sent through the C5 classifier as shown in Fig. 4. Classification occurs through matching patterns, to determine whether they demonstrate normal or abnormal behaviour. For the C5 to learn the pattern in the datasets, a classified dataset is required [55]. The dataset used by the C5 classifier is UNSW-NB15. The connections established will then be interpreted by the classifier and then get assigned to a specified class. In this case, the classifications are known attacks and normal traffic. The signatures of known attacks are then stored in the signature repository.

The packets sent from SIDS as normal traffic is now the input of AIDS which helps in finding the zero-day attack. AIDS is based on the learning of normal behavior which when implemented as a NIDS, helps in detecting abnormal behaviors in the network [44]. The classification in the proposed Hybrid Intrusion Detection System is binary which refers to the detection classification as 0 and 1 or normal and anomaly respectively. The dataset used to train the LTSM -RNN is ADFA-LD shown in Fig. 5 categorize data into normal and anomaly.

Signature generator:

In Fig. 4, generating signature from the packets requires attributes and features including signature repository. Wireshark can be used to extract features from detected malicious packets. The captured features are then compiled in CSV file and used in combination with the training set for the C5 model in stage one. The attributes collected are like that of UNSW-NB15. Wireshark is an open-source, commonly used network protocol analyzer that helps detect any suspicious packet entry, entering from an unreliable source. Wireshark is one of the most popular packet analyzers which is equipped with many features and can easily run on any platform. For signature generation and attribute extraction, Wireshark can perform various actions such as sniff, capture, log, and post-sniffing analysis. There are many paid applications and devices which help extract attributes and generate signatures. However, we consider Wireshark in this paper for a signature generation due to cost and ease of access [56].

4.2 Performance evaluation

An evaluation of performance can be used to determine whether algorithms, software, or systems are efficient in their operation. A performance metric is a measure of how well a system performs in order to evaluate the performance efficiency of the system. The evaluation of the performance by the classification models has been done in terms of standard performance metrics which are as follows: [15]

1) An accuracy measure is an indicator of how many instances are correctly classified among all instances within a dataset. An instance that was classified as a True Positive (TP) is defined as every instance that was classified correctly as a positive, plus every instance that was classified as a negative, divided by the total number of instances that were correctly classified as true positives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Where TP is true positive, TN is true negative, FP is false positive, and FN is False Negative.

2) As a performance metric, precision is defined as the ratio of true positives and the sum of true positive and false positive. It measures the fraction of instances that are predicted to be positive that is positive.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3) The recall measure refers to the ability of a model to identify all instances of a particular class or category correctly when it is evaluated in terms of its performance. It is the ratio of true positives and the sum of true positive and false negative. A true positive rate can also be called a recall rate or a sensitivity rate.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4) The F1 score is a performance metric that is often used in the assessment of the accuracy of binary classification models. It is a test of accuracy which is represented by the ratio of 2 times the multiplication of precision and recall and the sum of precision and recall. An F1 score is a measure of the harmonic mean of precision and recall, and it ranges from 0 to 1.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

5) The Receiver Operating Characteristic (ROC) is a performance evaluation technique that is used to analyze the trade-off between the true positive rate (TPR) and the false positive rate (FPR) in a data model with a variety of discrimination thresholds.

5 Results

There have been various scenarios in which the proposed method has been tested, and the results are discussed in this section based on the tests performed.

5.1 C5 classifier and dataset reduction

The dataset used in the decision tree model has been analysed and pre-processed before the integration of the data into the algorithm. To optimize the attack detection, the datasets will be pre-processed. The dataset comprised 45 variables, 4 of which were nominal and the remaining numerical. Furthermore, of the attributes in the dataset, 7 were categorical and the remaining were quantitative. The UNSW-NB15 training set has been reduced by eliminating redundant data. The training dataset had an attack column and ID column which were not of significance in the experiment and were dropped from the execution of the model. The dataset consisted of a column named 'label' which represented either attack or normal instances in binary. '1' represented attack instances whereas '0' represented normal instances. The order of the datasets was randomized, and the number of the normal instances was reduced to have a 1:1 proportion of attack:norm.

Also, the test dataset does not have the proto, state, and is_ftp_login columns which can be found in the training dataset. As a result, those columns were deleted from the training set. Moreover, the service column has missing instances that the C5 classifier cannot process when trained. Hence, this column has been dropped as well.

The R script to train the C5 model is as follows:

- C5_model ← C50(x = train_model[,−44], y = as.factor(train_model$attack_cat)) Where, UNSW-NB15_Trainingset has been assigned to train_model and to drop the column 'attack_cat' in predictors model 'train_model[,−44] has been used. Since, attack_cat is the 44th column in the training set. Similarly, the output, assigned as y is the attack_cat column.
- Summary (C5 model).

This script provides the detail of the model such as subtrees, size of the tree, attribute usage, and errors. From the summary, 1,693,409 instances were observed. 17% of which was an error. Similarly, the attribute usage of the datasets was observed. The attribute usage of UNSW-NB15 dataset in C5 classifier are shown in Table 6 with usage (%) and attribute parameters.

To train the training dataset, the following script was used,

- P1 ← predict (C5_model, test_data[,−44]) Here, P1 is the prediction model where test data is run through the previously generated C5 model.
- P1 This script runs the prediction model.

The performance metrics for the C5 classifier are shown in Table 7, which includes the TP, FP, precision, recall, and F1 score for both classes.

In comparison to [15] Al's research which ran C5 model to the UNSW-NB15 training set that had the number of instances reduced to 74,588, originally 152,148, produced the normal accuracy of 90.74% and the attack accuracy of 70.65%. The C5 model yielded better results than the proposed method in terms of accuracy [15]. However, [57] Als experiments yielded marginally superior accuracy through training and testing on combined UNSW-NB15 files with the ANN model yielding 99.26% average accuracy in binary classification and DNN yielding 99.22% accuracy in binary

Table 6 Attribute usage of UNSW-NB15 dataset in C5 classifier

Usage (%)	Attribute	Usage (%)	Attribute
71.58	sttl	5.69	sinpkt
57.10	smean	5.69	synack
48.21	sbytes	5.57	ct_dst_ltm
35.80	rate	4.49	ackdat
28.44	dinpkt	2.87	dmean
26.64	ct_srv_src	2.51	ct_src_dport_ltm
16.80	sloss	2.07	response_body_len
16.72	dbytes	1.98	dloss
13.99	trans_depth	0.83	dload
13.25	ct_dst_src_ltm	0.54	spkts
12.20	dpkts	0.30	dur
7.70	ct_dst_sport_ltm	0.25	sload
7.64	ct_state_ttl	0.21	stcpb
7.09	ct_srv_dst	0.21	dtcpb
0.12	ct_src_ltm	0.20	djit

Table 7 TP, FP, precision, Recall and F1 score for both classes with C5 classifier

Type	TP Rate	FP Rate	Precision	Recall	F1 score
Normal	0.977	0.060	0.942	0.921	0.902
Attack	0.969	0.103	0.903	0.891	0.897

classification. The dataset used in the experiments was a combined dataset of both training and testing sets that consisted of 2,540,044 packets [57]. Recent paper trains and tests UNSW-NB15 on two stacking models of which the first model has XGBoost and KNN as a base and Random Forest as a meta classifier and the second model is XGBoost, NN, KNN as a base and Random Forest as a meta classifier. Kabir et al. [58] achieved an accuracy of 93.62% with stack 1 and 92.76% with stack 2 which was higher than individual models such as XGBoost, NN, KNN, and RF [58]. The following Table 8 provides detailed accuracy by class for the UNSW-NB15 dataset was trained in the C4.5 model.

As seen from the table, the C5 outperforms the C4.5 classifier in almost every measure. In comparison to C5, TP Rate, FP Rate, Precision, Recall, F-Measure, MCC, ROC Area, PRC Area, and Class are not vastly different from C5, but C5 has proven to have greater performance in terms of these metrics.

5.2 LSTM-RNN

The ADFA-LD was evaluated and processed before the execution of LSTM. On the evaluation of the data, 6 types of attack were observed which are as follows:

There were 833 normal traces found in the training set and 4373 normal traces are found in the validation set. The attack data has been split into two sets, 70% of the attack data are used as a training set and 30% for validation. To achieve this, 7 folders of attack data have been used for the training set and the remaining 3 for validation. The LSTM model has 2 layers with 200 cells and the epoch was set between 50 and 5000 for training parse. Also, the learning rate was set to 0.001 to cover more data points.

Figure 7 represents the ROC curve of the LSTM-200. The LSTM classifiers yielded high accuracy and a low false alarm rate. The Area under ROC obtained for the experiment was 0.936 with a 17% false alarm rate (FAR). Table 9 represents the comparison of the performance of various models with the proposed LSTM. The specifications of the computer used include an Intel Core i711800H@2.3 GHz, 16 GB of RAM with NVIDIA GeForce RTX 3050 GPU running on 64-bit Windows 10. The following code was used as a reference to build the LSTM classifier:

http://github.com/ririhedou/systemCallAnomalyDetectionLSTM

In comparison to the results yielded through a CuDNNLSTM network which trained the ADFA-LD dataset in the paper by Borisaniya et al. [59] the proposed LSTM classifier slightly outperformed the bidirectional LSTM encoder as it achieved a TDR (True Detection rate) of 90% and FAR (False Alarm Rate) of 25% [59]. Also, Xie et al. [61] developed a System-call Behavioural Language based on a sensitivity-based LSTM model which achieved an AUC of 0.99 on test data and 0.93 on the unknown dataset [60]. The proposed LSTM model performs well in comparison to many ML and DL (deep learning) models and is on par with the most recent best-performing NIDS. However, the score achieved by the proposed model is significantly higher than most. Overall, the C5 classifier yielded an average true positive of 97.3% and an average false

Table 8 TP, FP, precision, recall and F1 score for both classes with C4.5 classifier

TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC area	PRC area	Class
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Normal
0.870	0.002	0.949	0.870	0.908	0.905	0.997	0.960	Reconnaissance
0.094	0.000	0.696	0.094	0.166	0.255	0.983	0.244	Backdoor
0.666	0.023	0.607	0.666	0.635	0.616	0.975	0.681	DoS
0.791	0.018	0.871	0.791	0.829	0.805	0.987	0.910	Exploits
0.097	0.000	0.868	0.097	0.175	0.290	0.981	0.277	Analysis
0.943	0.032	0.698	0.943	0.802	0.794	0.994	0.927	Fuzzers
0.591	0.000	0.867	0.591	0.703	0.716	0.999	0.682	Worms
0.836	0.001	0.847	0.836	0.842	0.841	0.998	0.866	Shellcode
0.992	0.001	0.997	0.992	0.995	0.993	1.000	0.999	Generic
0.929	0.006	0.934	0.929	0.925	0.922	0.996	0.953	Avg. weighted

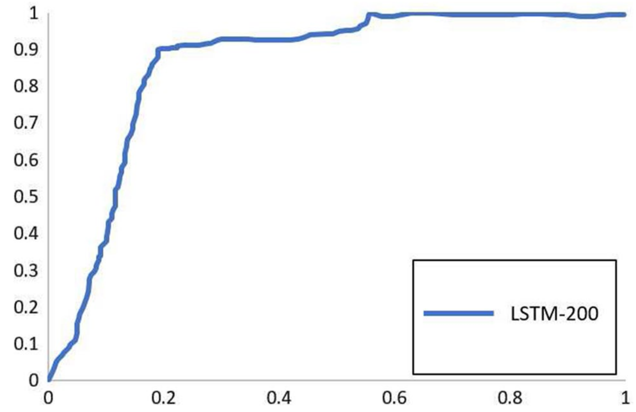


Fig. 7 False alarm rate vs detection rate for LSTM-200

Table 9 Trace counts and payloads of attack types in ADFA-LD

Attack	Payload	Trace count
Hydra-SSH	Brute force	176
Hydra-FTP	Brute force	162
Java-meterpreter	Java based meterpreter	124
Web shell	C100 Web shell	118
Add user	Add superuser	91
Meterpreter	Linux meterpreter	75

Table 10 Comparison table of HIDS

Models	AUC	Detection (%)	FAR
kMeans (k = 2) [59]	0.481	82.86	N/A
Bidirectional CUDNNLSTM 200 [60]	0.86	90	25%
Sensitivity-Based LSTM [61]	0.99 (known attacks)	89 (average)	N/A
	0.93 (unknown attacks)	89 (average)	N/A
XGBoost [62]	N/A	95	0.01%
Hybrid Classifier [63]	94%	99.7	1.29%
Spark ML + Conv-LSTM [30]	N/A	97	0.71%
Proposed LSTM	0.936	90	17%

positive of 8% which is among the class leaders in signature-based intrusion detection systems. Also, the proposed LSTM model yielded a detection rate of 90% maintaining a very low false alarm rate of 17%. Both the stages of the HIDS have displayed class-leading performance metrics as shown in Table 10 which compares the result of the proposed IDSs with some of the best-performing IDS models.

6 Conclusion and future works

Real-time packet testing has been identified as a future research project. To test the effectiveness and accuracy of the proposed model, it needs to be subjected to real-time packets. These packets should contain various attack types as well as should represent benign network traffic. The main intention of the real-time network testing is to verify the anomalous packets, extract the malware attribute and feed it to the signature repository to train the C5 model in the signature-based intrusion detection stage. After this, upon the execution of the same attack, the SIDS must trigger the alert before reaching the anomaly-based intrusion detection stage. Attack signatures that have been found externally can also be added to the repository to train the SID. Moreover, the performance of the HIDS model collectively needs to be assessed in terms of accuracy, detection rate, and false alarm rate. The retention of signatures and the execution of those signatures in the SIDS stage needs to be assessed as well. This helps measure the self-healing ability of the proposed model. Developing signature generation techniques requires more focus in future work, as there are several novel methods and devices to extract attributes from anomalous packets more efficiently. Appropriate feature extraction/attribute extraction methods should be evaluated, and a method of signature extraction needs to be selected those complements and is cohesive with the proposed model. The paper does not address obfuscation techniques and methods to mitigate the inability to detect obfuscated packets. As this poses a threat to the proposed model, there arises an opportunity to act upon the mitigation of such threats. In order to further research, computational power and time can be taken into account for the distribution of the proposed model in real life. This will take into account the efficiency, affordability, and practicality of the model in a network.

Author contributions SK contributed to acquisition and analysis of data, conception and design of methodology, writing original draft, SB contributed towards review, supervision, and editing. ST contributed towards conception and design of methodology, supervision, review. JS contributed towards conception and design of methodology, final edits and validation.

Data availability The datasets analysed during the current study are available at the following link: https://research.unsw.edu.au/projects/unsw-nb15-dataset.

Declarations

Competing interests The authors declare that they have no competing of interests.

References

1. Alsamiri J, Alsubhi K. Internet of things cyber attacks detection using machine learning. Int J Adv Comput Sci Appl. 2019;10(12):628–34.
2. Zamani M, Movahedi M. Machine learning techniques for intrusion detection. 2013. arXiv preprint arXiv:1312.2177
3. Liao H-J, Lin C-HR, Lin Y-C, Tung K-Y. Intrusion detection system: a comprehensive review. J Netw Comput Appl. 2013;36(1):16–24.
4. Al-Qatf M, Lasheng Y, Al-Habib M, Al-Sabahi K. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. IEEE Access. 2018;6:52843–56.
5. Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. 2017;5:21954–61.
6. Min E, Long J, Liu Q, Cui J, Chen W. TR-IDS: anomaly-based intrusion detection through text-convolutional neural network and random forest. Secur Commun Netw. 2018. https://doi.org/10.1155/2018/4943509.
7. Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans Emerg Telecommun Technol. 2021;32(1):4150.
8. Sarker IH, Kayes A, Badsha S, Alqahtani H, Watters P, Ng A. Cybersecurity data science: an overview from machine learning perspective. J Big data. 2020;7:1–29.
9. Garcia-Teodoro P, Diaz-Verdejo J, Maciá-Fernández G, Vázquez E. Anomaly-based network intrusion detection: techniques, systems and challenges. Comput Secur. 2009;28(1–2):18–28.
10. Kotecha K, Verma R, Rao PV, Prasad P, Mishra VK, Badal T, Jain D, Garg D, Sharma S. Enhanced network intrusion detection system. Sensors. 2021;21(23):7835. https://doi.org/10.3390/s21237835.
11. Imrana Y, Xiang Y, Ali L, Abdul-Rauf Z. A bidirectional LSTM deep learning approach for intrusion detection. Expert Syst Appl. 2021;185: 115524.
12. Kaur S, Singh M. Hybrid intrusion detection and signature generation using deep recurrent neural networks. Neural Comput Appl. 2020;32:7859–77.
13. Sarhan M, Layeghy S, Portmann M. Towards a standard feature set for network intrusion detection system datasets. Mobile Netw Appl. 2022. https://doi.org/10.1007/s11036-021-01843-0.
14. Kumar V, Das AK, Sinha D. Statistical analysis of the UNSW-NB15 dataset for intrusion detection. In: Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019, Springer. 2020; pp. 279–94.
15. Thapa N, Liu Z, Kc DB, Gokaraju B, Roy K. Comparison of machine learning and deep learning models for network intrusion detection systems. Future Internet. 2020;12(10):167.
16. Gamage S, Samarabandu J. Deep learning methods in network intrusion detection: a survey and an objective comparison. J Netw Comput Appl. 2020;169: 102767.

17. Elsadig M, Abdullah A. Biological inspired intrusion prevention and self-healing system for network security based on danger theory. Int J Video Image Process Netw Secur. 2009;9(9):16–28.
18. Hajisalem V, Babaie S. A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. Comput Netw. 2018;136:37–50.
19. Gómez J, Gil C, Padilla N, Baños R, Jiménez C. Design of a snort-based hybrid intrusion detection system. In: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living: 10th International Work-Conference on Artificial Neural Networks, IWANN 2009 Workshops, Salamanca, Spain, June 10–12, 2009. Proceedings, Part II 10, Springer. 2009; pp. 515–22.
20. Chegini H, Beltran F, Mahanti A. Designing and developing a weed detection model for California thistle. ACM Trans Internet Technol. 2022. https://doi.org/10.1145/3544491.
21. Sial A, Singh A, Mahanti A. Detecting anomalous energy consumption using contextual analysis of smart meter data. Wirel Netw. 2021;27:4275–92.
22. Jain DK, Kotecha K, Pandya S, Reddy SS, Varadarajan V, Mahanti A, et al. Hybrid deep neural network for handling data imbalance in precursor MicroRNA. Front Public Health. 2021;9:2161.
23. Dhatrak A, Gong M, Naha R, Mahanti A. Secure IoT data using blockchain. In: Verma A, Verma P, Farhaoui Y, Lv Z, editors. Emerging real-world applications of internet of things. Boca Raton: CRC Press; 2022. p. 1–20.
24. Meng W, Tischhauser EW, Wang Q, Wang Y, Han J. When intrusion detection meets blockchain technology: a review. IEEE Access. 2018;6:10179–88.
25. Tesfahun A, Bhaskari DL. Intrusion detection using random forests classifier with smote and feature reduction. In: 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, IEEE. 2013; pp. 127–32.
26. Chitrakar R, Chuanhe H. Anomaly detection using support vector machine classification with k-medoids clustering. In: 2012 Third Asian Himalayas International Conference on Internet, IEEE. 2012; pp. 1–5.
27. Ahmad I, Basheri M, Iqbal MJ, Rahim A. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access. 2018;6:33789–95.
28. Aminanto E, Kim K. Deep learning in intrusion detection system: An overview. In: 2016 International Research Conference on Engineering and Technology (2016 IRCET). Higher Education Forum. 2016.
29. Khan MA, Karim MR, Kim Y. A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. Symmetry. 2019;11(4):583.
30. Aloseel A, Al-Rubaye S, Zolotas A, Shaw C. Attack-detection architectural framework based on anomalous patterns of system performance and resource utilization-part II. IEEE Access. 2021;9:87611–29.
31. Kim G, Lee S, Kim S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst Appl. 2014;41(4):1690–700.
32. Kim J, Kim J, Thu HLT, Kim H. Long short term memory recurrent neural network classifier for intrusion detection. In: 2016 International Conference on Platform Technology and Service (PlatCon), IEEE. 2016; pp. 1–5.
33. Rizvi S, Labrador G, Guyan M, Savan J. Advocating for hybrid intrusion detection prevention system and framework improvement. Proc Comput Sci. 2016;95:369–74.
34. Aydin M, Zaim AH, Ceylan KG. A hybrid intrusion detection system design for computer network security. Comput Electr Eng. 2009;35(3):517–26.
35. Degeler V, French R, Jones K. Self-healing intrusion detection system concept. 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), IEEE. 2016; pp. 351–56.
36. Bovenzi G, Aceto G, Ciuonzo D, Persico V, Pescapé A. A hierarchical hybrid intrusion detection approach in IoT scenarios. In: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE. 2020; pp. 1–7.
37. Sohi SM, Seifert J-P, Ganji F. RNNIDS: enhancing network intrusion detection systems through deep learning. Comput Secur. 2021;102: 102151.
38. Creech G, Hu J. Generation of a new IDS test dataset: Time to retire the KDD collection. In: 2013 IEEE Wireless Communications and Networking Conference (WCNC), IEEE. 2013; pp. 4487–92.
39. Layeghy S, Gallagher M, Portmann M. Benchmarking the benchmark–analysis of synthetic nids datasets. 2021. arXiv preprint arXiv: 2104.09029.
40. Chew YJ, Lee N, Ooi SY, Wong K-S, Pang YH. Benchmarking full version of GureKDDCup, UNSW-NB15, and CIDDS-001 NIDS datasets using rolling-origin resampling. Inf Secur J Glob Perspect. 2022;31(5):544–65.
41. Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. IEEE Access. 2019;7:41525–50.
42. Bachar A, El Makhfi N, Bannay OE. Towards a behavioral network intrusion detection system based on the SVM model. In: 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), IEEE. 2020; pp. 1–7.
43. Kim J, Kim H, et al. An effective intrusion detection classifier using long short-term memory with gradient descent optimization. In: 2017 International Conference on Platform Technology and Service (PlatCon), IEEE. 2017; pp. 1–6.
44. Aldweesh A, Derhab A, Emam AZ. Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl-Based Syst. 2020;189: 105124.
45. Otoum Y, Nayak A. As-ids: Anomaly and signature based ids for the internet of things. J Netw Syst Manag. 2021;29:1–26.
46. Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M. Deep recurrent neural network for intrusion detection in sdn-based networks. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), IEEE. 2018; pp. 202–6.
47. Belavagi MC, Muniyal B. Performance evaluation of supervised machine learning algorithms for intrusion detection. Proc Comput Sci. 2016;89:117–23.
48. Althubiti SA, Jones EM, Roy K. LSTM for anomaly-based network intrusion detection. In: 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), IEEE. 2018; pp. 1–3.

49. Ibrahim HE, Badr SM, Shaheen MA. Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems 2012. arXiv preprint arXiv:1210.7650.
50. Naidu RCA, Avadhani P. A comparison of data mining techniques for intrusion detection. In: 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), IEEE. 2012; pp. 41–4.
51. Nasr MM, Shaaban EM, Gabr MI. Comparative study: classification algorithms before and after using feature selection techniques. Int J. 2017. https://doi.org/10.23956/ijarcsse/V7I2/01212.
52. Moustafa N, Hu J, Slay J. A holistic review of network anomaly detection systems: a comprehensive survey. J Netw Comput Appl. 2019;128:33–55.
53. Reis M, Paula F, Fernandes D, Geus P. A hybrid ids architecture based on the immune system. In: Anais do II Workshop em Segurança de Sistemas Computacionais, SBC. 2002; pp. 127–34.
54. Hwang K, Cai M, Chen Y, Qin M. Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. IEEE Trans Depend Secur Comput. 2007;4(1):41–55.
55. Kumar M, Hanumanthappa M, Kumar TS. Intrusion detection system using decision tree algorithm. In: 2012 IEEE 14th International Conference on Communication Technology, IEEE. 2012; pp. 629–34.
56. Banerjee U, Vashishtha A, Saxena M. Evaluation of the capabilities of Wireshark as a tool for intrusion detection. Int J Comput Appl. 2010;6(7):1–5.
57. Aleesa A, Younis M, Mohammed AA, Sahar N. Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques. J Eng Sci Technol. 2021;16(1):711–27.
58. Kabir MH, Rajib MS, Rahman ASMT, Rahman MM, Dey SK. Network intrusion detection using unsw-nb15 dataset: Stacking machine learning based approach. In: 2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), IEEE. 2022; pp. 1–6.
59. Borisaniya B, Patel D, et al. Evaluation of modified vector space representation using adfa-ld and adfa-wd datasets. J Inf Secur. 2015;6(03):250.
60. Chawla A, Jacob P, Lee B, Fallon S. Bidirectional LSTM autoencoder for sequence based anomaly detection in cyber security. Int J Simul Syst Sci Technol. 2019. https://doi.org/10.5013/IJSSST.a.20.05.07.
61. Xie W, Xu S, Zou S, Xi J. A system-call behavior language system for malware detection using a sensitivity-based LSTM model. In: Proceedings of the 3rd International Conference on Computer Science and Software Engineering, 2020; pp. 112–8.
62. Kotecha K, Verma R, Rao PV, Prasad P, Mishra VK, Badal T, Jain D, Garg D, Sharma S. Enhanced network intrusion detection system. Sensors. 2021. https://doi.org/10.3390/s21237835.
63. Samunnisa K, Kumar GSV, Madhavi K. Intrusion detection system in distributed cloud computing: hybrid clustering and classification methods. Meas: Sens. 2023;25: 100612.