Discover Computing

Research

Enhancing network intrusion detection systems with combined network and host traffic features using deep learning: deep learning and IoT perspective

Estabraq Saleem Abduljabbar Alars[1] · **Sefer Kurnaz**[1]

Received: 29 July 2024 / Accepted: 18 October 2024
Published online: 04 November 2024

Abstract
Network security is a key concern in today's linked world as cyber threats grow more sophisticated and ubiquitous. Traditional Network Intrusion Detection Systems (NIDS) generally fall short owing to their dependence on predetermined signatures and restricted detection scope, exposing substantial gaps in efficiently recognizing new and unanticipated intrusions. This research tackles these difficulties by merging network and host traffic data with sophisticated deep learning algorithms to boost NIDS performance. Utilizing the Network Intrusion Detection dataset, which comprises multiple intrusion scenarios replicated in a military network context, our technique involves painstaking data collection, preprocessing, and feature extraction. We employed a convolutional neural network (CNN) to assess these data, applying rigorous feature selection and dimensionality reduction to enhance model performance. The findings reveal that our deep learning-based NIDS achieves an amazing detection accuracy of 98.5%, exceeding current approaches and successfully resolving real-world cybersecurity problems. This complete approach not only develops NIDS technology but also provides a practical solution for boosting network security across many applications, therefore contributing to the development of intrusion detection systems.

Keywords Deep learning · Network intrusion detection system · Cybersecurity · Feature extraction · Convolutional neural network · Network traffic analysis · Host traffic features

1 Introduction

In today's world of cyber security, a solid network intrusion detection system (NIDS) is one of the most crucial components of a network security system. NIDS are meant to record and analyze traffic on a network, checking for unwanted traffic and a plethora of other risks. These systems are particularly crucial in defending networked systems and provide ways of spotting various forms of danger in the cyber realm and preserving information [1]. This is because the threats are developing and getting more sophisticated, and there is a need for better and more efficient NIDS [2].

Traditional NIDS generally depend on two approaches: The two common techniques that have been developed are the signature-based detection and the anomaly-based detection [3]. Although signature-based systems are successful in identifying existing dangers, they are not particularly good at detecting new threats, such as zero-day assaults [4]. While anomaly based systems may discover novel threats, they are frequently accompanied by high false positive rates that

✉ Estabraq Saleem Abduljabbar Alars, Estabraq.alars@ogr.altinbas.edu.tr | [1]Department of Electrical and Computer Engineering, Altinbas University, 34000 Istanbul, Turkey.

overload the security team with multiple alerts [5]. Furthermore, classic NIDS have a very restricted range of coverage, which is largely geared at the analysis of network data while entirely disregarding host traffic. This shortcoming affects their power to recognize assaults that may target both the network and host properties [6].

That is why there is a significant possibility for future improvement, taking into mind the indicated above limits. A combination of the network traffic analysis with the host traffic analysis might enhance the detection rates by a substantial margin and at the same time decrease the false positives [7, 8].

The shortcomings of traditional NIDS necessitate a more thorough strategy to achieve successful detection of new and recognized threats with a limited amount of false alarms. Current solutions do not make full use of the information that may be acquired from host traffic, leaving alternative potential for attacks undiscovered. This study covers the gap in research that has advocated for the creation of a hybrid network traffic and host traffic intrusion detection system.

This study intends to enhance the performance of NIDS by the incorporation of deep learning methods in assessing both network and host traffic characteristics. Traditional systems have their shortcomings, and deep learning, which is capable of analyzing huge data and identifying complicated patterns, may become the solution to these difficulties. The suggested strategy is to merge the network and host data, and in this manner, raise the detection rate and lower the number of false alarms.

- To design a hybrid NIDS that incorporates network and host traffic aspects.
- To employ deep learning algorithms to assess these combined characteristics for anomaly detection.
- To compare the performance of the proposed system versus standard NIDS techniques.

The practical consequence of this hybrid NIDS is that it has practical use in the world of cybersecurity. This system may give the tools to boost the detection of threats and limit false alarms, thereby maintaining the security of data, the dependability of important facilities, and the continuity of operations. The application of deep learning enhances the capacity of the system in spotting sophisticated attack patterns, which gives a strong response to the present security concerns.

The future work may be expanded to obtain additional traffic data to test the suggested system for a broader variety of traffic circumstances and to try out various machine learning algorithms to better the performance of the system. Further, the integration of real-time adaptive learning might increase the NIDS's potential to react to new threats and hence maintain high detection rates in the long term.

The following is a thorough article on strengthening NIDS via the combination of network and host traffic features and deep learning algorithms. By means of such an integration, the study is projected to increase the efficiency, efficacy, and overall dependability of IDSs while overcoming the weaknesses of traditional NIDSs and opening the way for future improvements in the domain of information security.

2 Literature review

2.1 Network intrusion detection systems

Network intrusion detection systems, or NIDS for short, are crucial tools in the cyber security space since they are designed to identify and thwart illegal activity occurring inside a network. The two primary categories of automated techniques used in classic NIDS technologies are signature-based and anomaly-based systems [9]. Conversely, SNIDS operates by identifying network traffic in conjunction with industry-recognized threat guidelines. The effectiveness of these security measures is greatest when the danger is well-known and predictable; however, they are not particularly helpful in spotting new threats [10, 11]. A figure -1 illustrate the network intrusion detection system (NIDS).

However, anomaly-based network intrusion detection systems establish a baseline for network activity, and any deviation from this is seen as a danger. The efficacy of the system is decreased even when they catch more unknown threats since they lead to more false positive warnings, which strain the security analysts [12]. The goal is to minimize false alarm levels while simultaneously achieving a high degree of accuracy in detection, as stated by Carter and made unclear in the application form [13] as shown in Fig. 1.

The second important method in NIDS for completing threat characteristic analysis and, thus [14], quickly and easily detecting intricate assault patterns is pattern matching. The development of extensive and effective pattern matching

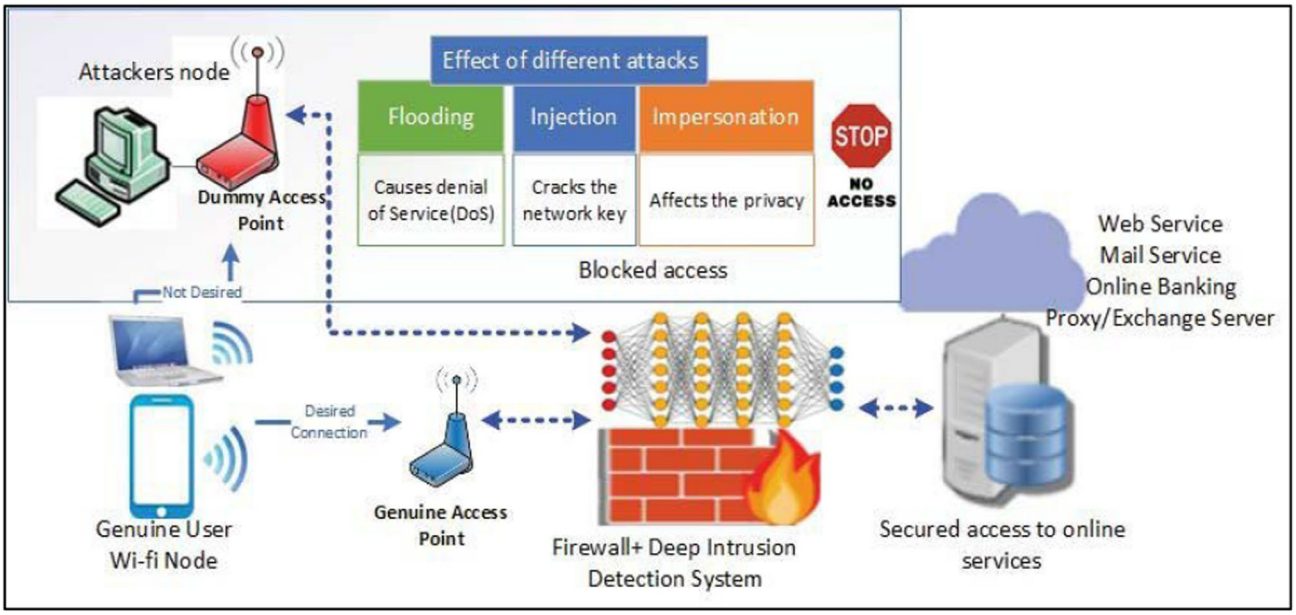

Fig. 1 Network intrusion detection system (NIDS) [11]

tools has also contributed to advancements in this sector, improving NIDS responsiveness [15]. A lot of network traffic has also led to the presentation of new high-performance NIDS structures, which enhance their ability to identify intrusions in real time [16, 17].

2.2 Host and network traffic features

There will be methods by which the flow of NIDS will be significantly increased by using both hosts and network traffic [18]. While system calls, user activity log files, and application logs are included in host traffic data, packet headers, payload data [19], and flow elements are included in network traffic data [20, 21].

Numerous elements increase the accuracy rate of intrusion detection, according to research. To enhance the detection capabilities of a network intrusion detection system, for instance, it has been discovered that the feature selection technique places more emphasis on a methodical selection process [22]. In a similar vein, fresh statistical flow characteristics tailored to IoT applications have been created to further improve network traffic security [23]. Figure 2 show the example of Overview of the feature extraction process.

Since feature extraction is primarily responsible for highlighting the correct data structures, it is another crucial step [24, 25]. It has been discovered that methods of feature abstraction, including behavioral-based feature abstraction [26], are very helpful in distinguishing between normal and abnormal traffic[27, 28]. Additional methods that improve the monitoring of growing volumes of large-scale network traffic include dynamic feature analysis [29].

2.3 Deep learning in NIDS

When it comes to NIDS, deep learning techniques are useful and successful since they aid in the identification of intricate and dynamic risks [30]. Once again, a great deal of adoption of recurrent neural networks (RNNs) and long-short-term memory (LSTM) networks has occurred in this field because of the temporal dependencies in network traffic [31].

Research has shown that the use of deep learning models may provide significant improvements in detection accuracy. For example, better outcomes in identifying different types of assaults were found in a study that used deep learning to apply NIDS [32]. IDSs that are very accurate and successful have been produced by reevaluating the integration of several deep learning architectures [33, 34]. A Machine learning and deep learning-based intrusion detection system discuss in Fig. 3.

Additionally, in order to stop adversarial impacts on NIDS [35, 36], adversarial machine learning techniques are being researched [5]. Additionally, by using these tactics, NIDS is more reliable in thwarting contemporary assault evasion

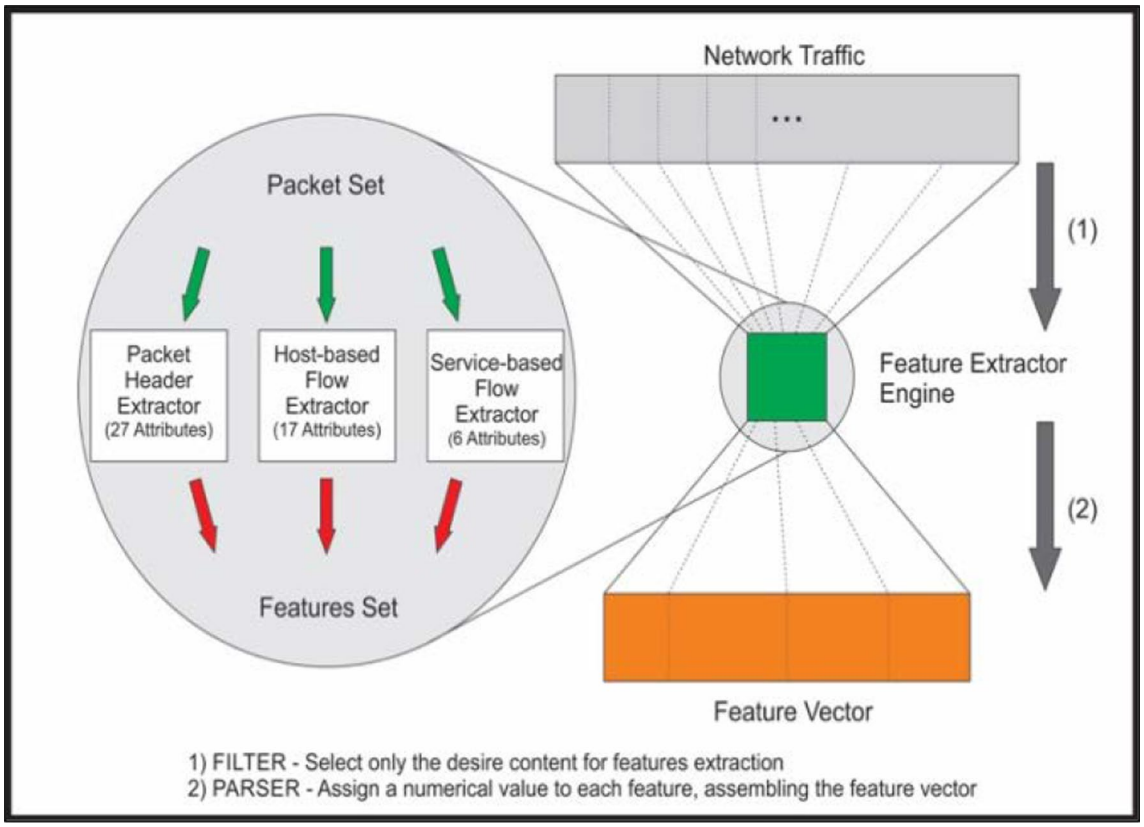

Fig. 2 Overview of the feature extraction process [28]

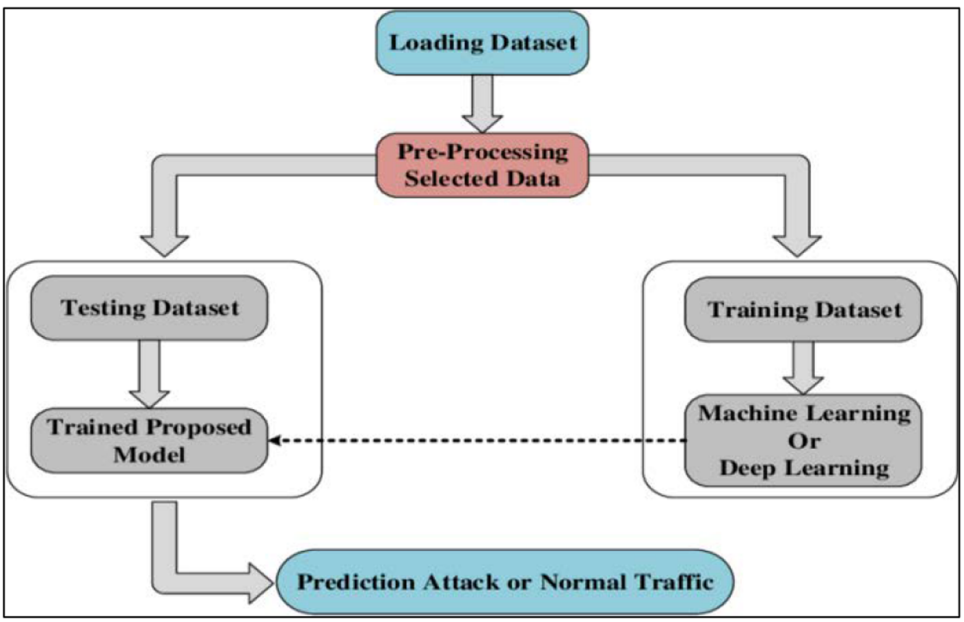

Fig. 3 Machine learning and deep learning-based intrusion detection system [34]

strategies [37]. It may be possible to develop novel, adaptable, and efficient solutions by using deep learning in conjunction with the traditional NIDS frameworks [38].

this paper aims to develop a more comprehensive and effective NIDS by combining network and host traffic features with deep learning techniques, addressing the limitations of current systems and enhancing overall cybersecurity.

2.3.1 IoT security challenges

The proliferation of IoT devices has rendered networks more susceptible to security attacks. Intrusion Detection Systems (IDS) effectively mitigate these hazards by promptly detecting and responding to intrusions in real time. Nevertheless, the dynamic and evolving nature of IoT settings poses specific obstacles to IDS, necessitating the implementation of advanced and adaptable solutions.

2.3.2 Recent advances in IDS for IoT

Several studies have made noteworthy contributions to enhancing intrusion detection systems (IDS) for Internet of Things (IoT) environments. A framework was proposed to enhance the ability of NIDS to withstand adversarial attacks by using visual analytics to improve the detection rate [40]. Likewise, a novel two-phase defensive approach was suggested to prevent untargeted white-box optimization adversarial assaults, which boosts the IDS's resilience by a substantial proportion [41].

To boost the IDS's resilience and robustness, a sequential deep learning architecture was proposed that combines the newest deep learning approaches to enhance the detection rates [42]. A defense system was created that is effective against adversarial machine learning attacks targeting IDS in particular, exhibiting increased defensive performance [43].

A fundamental framework was provided to increase the adversarial resilience of deep learning-based IDS, targeting computational efficiency and real-world application [44]. The use of stochastic gradient descent in intrusion detection in wireless sensor networks was examined, demonstrating that machine learning may be utilized to increase attack detection in constrained scenarios [45].

A deep learning-based IDS employing a chaotic optimization strategy was presented to increase the detection rate [46]. A comprehensive literature analysis on effective IDS models was completed, highlighting varied techniques and their efficiency [47].

Focus was placed on the combination of tabular and text-based characteristics for network intrusion detection, proposing a unique notion of utilizing different data types [48]. A meta-analysis and systematic review of anomaly-based IDS was conducted, reviewing detection algorithms, datasets, and validation procedures [49].

2.3.3 Recent literature on IoT-specific IDS

The latest works have underlined the requirement for IoT-specific techniques [50]. Detailed approaches for system intrusion detection and prevention, notably in IoT environments, were developed [51]. A technique was provided for detecting corrupted training datasets for machine learning-based IDS, thereby boosting the IDS's dependability in diverse IoT situations [52].

AI-enabled IDS was suggested in the context of Industrial IoT, reviewing advancements in network security based on deep learning [53]. A hypergraph-based machine learning ensemble IDS was developed, introducing a novel paradigm for enhancing detection [54]. The merging of the Bat algorithm with the Residue Number System for feature selection in IDS was proposed, boosting detection performance [55].

A deep learning-based ensemble architecture for IDS dubbed ENIDS was introduced to increase detection performance in complex IoT environments [56]. An adversarial-resistant IDS, ARGAN-IDS, leverages generative adversarial networks to prevent adversarial assaults [57].

Representation learning methods were evaluated to boost IDS resilience, emphasizing novel approaches to increase the detection rate [58]. A novel CNN-IDS model was proposed to enhance the effectiveness of intrusion detection, highlighting the significant influence of CNN structures on IDS [59].

2.3.4 Major findings and comparison

In addition to these contributions, several key studies have provided significant insights into the evolution of IDS technologies:

- A novel intrusion detection and classification system for IoT traffic with increased data engineering has been introduced to illustrate improvements in managing IoT traffic [60].

- An examination of explainable IDS based on Convolutional Neural Networks utilizing Shapley Additive Explanations focuses on interpretability [61].
- An intelligent method for identifying and classifying harmful URLs that are beneficial in web-based assaults in IoT has been suggested [62].
- A deep convolutional neural network for detecting and classifying cyber-attacks in IoT communication provides a thorough solution to IoT security [63].
- Additionally, a cost-efficient hybrid learning strategy for recognizing cross-site scripting assaults emphasizes cost and accuracy [64].

To provide a clear overview of the state-of-the-art models and their methodologies, the following comparative table summarizes key aspects of surveyed papers as given in Table 1:

The expanded literature review underscores the advancements in IDS methodologies, particularly in the context of IoT environments. The comparative analysis highlights significant progress in enhancing IDS robustness, accuracy, and efficiency. Incorporating these insights provides a comprehensive understanding of current trends and challenges in IDS for IoT systems.

3 Methodlogy

This study offers a novel concept for enhancing NIDS by using deep learning techniques to bridge the heterogeneous data related to host traffic and the network. The research approach comprises the main crucial phases. Firstly, we collect large datasets comprising atomic host level activity as well as flow traffic. Subsequently, the characteristics of the discovered attributes from both data sets undergo pre-processing. After that, we modify those characteristics in order to feed them into the various deep learning algorithms. Our suggested deep learning architecture, which is suited for intrusion detection, is the next layer. It is trained using the feature set previously described. The efficacy of the model is then assessed using multiple performance metrics, such as accuracy, precision, recall, and F1-score, to guarantee the model's ability to identify various kinds of cyberthreats. As a result, this all-encompassing strategy may improve detection accuracy and reduce false positives, which are the primary flaw in conventional NIDS. A proposed system illustrate in Fig. 4.

3.1 Data collection

The dataset employed in the current research is the Network Intrusion Detection dataset [39], a widely-used dataset aiming at enhancing NIDS by merging both network and host traffic using deep learning. This dataset is a typical US Air Force Local Area Network (LAN), which contains all the TCP/IP dump data that covers a range of network activities and realistic assaults as well as routine traffic. Specifically, the dataset comprises raw TCP/IP connection logs, each of which is stored in a 100-byte format. The records are thorough with 41 qualities, where two are ordinal, two are ratio, and 37 are dichotomous, which offer a full description of a network connection. The data are classified either as benign or as belonging to one of seven separate categories of network assaults, which suggests that there is a large diversity of potential security dangers.

The data collection technique included two key sources: network traffic data and host traffic data. Network traffic data was gathered via TCP/IP dumps, which provide information about packets, including the source and destination IP addresses, protocol, and payload. At the same time, host traffic data was acquired from system logs and user activity records in the simulated LAN environment. This data covers user activities, system activities, and application activities, giving a holistic picture of the network and host activities. The combination of the network and host traffic data is vital to our deep learning model as it offers a more comprehensive picture of intrusions and abnormalities.

As for the dataset and the parameters used in the deep learning models, the Network Intrusion Detection dataset is downloaded from the official source or other data repositories, but access limited to registration or a request. In model training and assessment, all 41 characteristics of the dataset were employed. This was followed by normalization of numerical features, encoding of categorical characteristics, and separation of the data set into training and test data sets to increase the performance of the model. The deep learning models were configured to train at a learning rate of 0. The models were trained using 001, a batch size of 64, and for 50 epochs as shown in Fig. 5.

Table 1 Comparative summary table

Study	Methodology	Performance metrics	Computational efficiency	Key findings
[40]	Visual interpretation, adversarial robustness	Accuracy, robustness	High	Improved robustness using visual techniques
[41]	Two-phase defense strategy	Robustness, accuracy	Moderate	Effective against untargeted white-box attacks
[42]	Sequential deep learning framework	Resilience, accuracy	High	Enhanced resilience and detection performance
[43]	Adversarial machine learning defense	Defense effectiveness	Moderate	Improved defense against adversarial attacks
[44]	Simple framework for adversarial robustness	Robustness, efficiency	High	Simplified approach for enhanced robustness
[45]	Stochastic gradient descent	Detection accuracy	Low	Effective in wireless sensor networks
[46]	Dugat-LSTM with chaotic optimization	Detection accuracy	Moderate	Enhanced accuracy using chaotic optimization
[47]	Systematic literature review	Comparative metrics	Not applicable	Comprehensive review of IDS models
[48]	Hybrid approach: tabular and text-based	Accuracy, flexibility	High	Integration of diverse features
[49]	Meta-analysis of anomaly detection	Detection methods	Moderate	Comprehensive review of anomaly detection methods

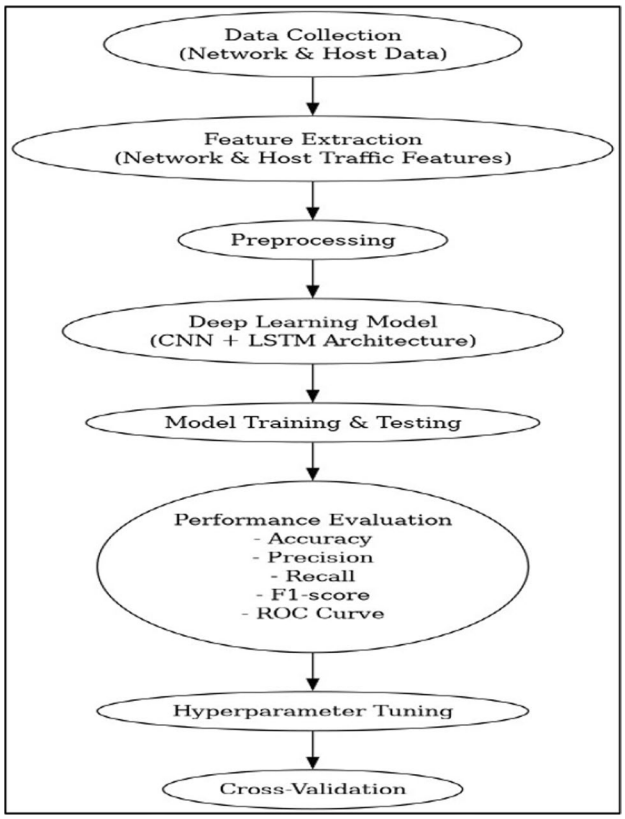

Fig. 4 Propose system for network intrusion detection systems with combined network and host traffic features using deep learning

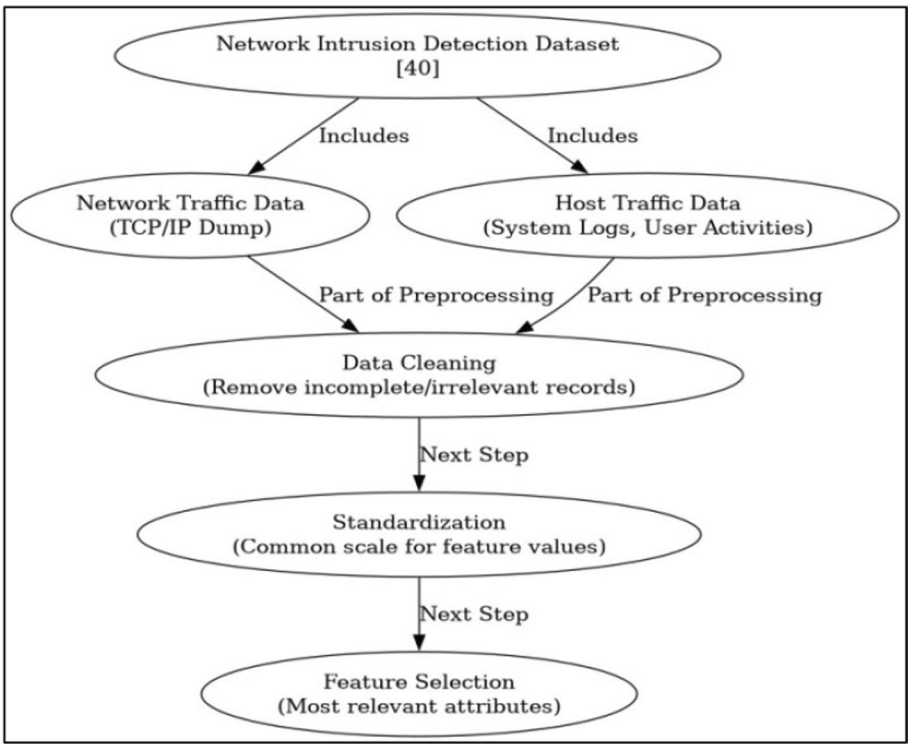

Fig. 5 Dataset overview [39]

3.2 Data preprocessing

In particular, data transformation and cleaning are crucial to our system because they provide us with a suitable and trustworthy dataset that we can use to train our deep learning model. Previously, we handled missing data by replacing them with mean or median values if their percentage was smaller than the total features, or by removing such entries. Subsequently, the dataset underwent normalization to ensure that every feature was of the same magnitude. This is an essential step in ensuring the effectiveness of deep learning algorithms. To get the majority of the feature values within a range of a normalized value of 0 and 1, we used rescaled formulas and normalization on the feature values. Additionally, the one-hot encoding approach was employed to convert category information into numerical characteristics that could be used by models.

The Network Intrusion Detection dataset [39] was used in this study. This dataset includes a wide variety of intrusions simulated in a military network environment, with each connection record containing 41 features. The dataset was preprocessed as follows:

- Data Cleaning: missing values were handled, and irrelevant features were removed.
- Normalization: feature scaling was applied to ensure all features contributed equally to the learning process.
- Label Encoding: categorical features were converted into numerical format using one-hot encoding.
- Train-Test Split: the dataset was split into training (70%), validation (15%), and testing (15%) sets to evaluate the model performance.

3.3 Feature extraction

A key component of the effective Network Intrusion Detection System (NIDS) design process is feature extraction. To better monitor anomalous behavior, a collection of additional characteristics was built in this study by carefully gathering comprehensive network and host traffic data in relation to known features.

Traffic characteristics were derived from traffic traces since the network traffic was taken from unprocessed TCP/IP dump data. Each network connection was comprised of a source or destination IP address and was broken down into a sequence of TCP packets. 41 characteristics were used to characterize each connection. Therefore, the features were divided into the following categories: fundamental features, which allowed for the creation of properties such as duration, protocol type, service, and flag; content features that provide information about data content, such as the quantity of unsuccessful connection attempts and "hot" indicators that characterize actions akin to file access or shell prompts; time-based traffic features that recorded temporal aspects of network traffic, such as the quantity of connections to comparable hosts or services in a specific amount of time; and, lastly, host-based traffic features.

System logs and user activity records are sources of information used in the host traffic features. System calls that recorded the order and frequency of system calls made by the host's processes; user actions that preserved information about log-ins, file operations, and changes to user privileges; and applications that tracked the actions within specific applications, like database transactions or Web server requests, were some of the features.

The integrated network and host traffic characteristics had a large dimensionality; therefore, feature space reduction and feature selection were done to increase the computational efficiency and performance of the model. There were characteristics with a high correlation coefficient, and the ones least significant to the identification of intrusions were eliminated using the correlation analysis approach. The dataset's dimensions were reduced while attempting to preserve as much variation as possible via the use of principal component analysis (PCA). The outcome of PCA was a collection of orthogonal components representing the dataset's characteristics. In addition, characteristics that are less crucial to the model at each stage were eliminated by applying model constructs using recursive feature elimination (RFE).

3.4 Deep learning model

The deep learning model's architecture was designed to exploit the network's layered structure and incoming traffic data. We used both convolutional neural networks (CNNs) and long short-term memory (LSTM) models to take advantage of their individual capabilities in analyzing image-based and sequential data, respectively. CNNs are excellent at extracting spatial aspects of the data. In terms of feature representation, the current model used CNN layers, which are extensively

used in deep learning, to effectively learn and extract abstract characteristics from the raw input. Specifically, LSTM networks, which are a subtype of recurrent neural networks, are utilized to capture temporal relationships and sequential behavior in the network traffic data.

The purposeful use of both convolutional neural networks (CNNs) and long-short-term memory (LSTMs) was motivated by the following justifications: CNNs are particularly effective at learning topological properties in data, which is crucial for spotting complex patterns in the network traffic. LSTMs have a remarkable quality of processing sequential input and learning the connections between the sequences, which is highly beneficial for learning the dynamics of the events in the network. The hybrid model utilizes the features from both CNNs and LSTMs to develop a powerful framework that can efficiently recognize intrusions with the aid of both spatial and temporal properties. This architecture is capable of readily accepting the high dimensional and sequential properties of the network and host traffic data, thereby enhancing the detection rate and lowering false positives inside the model.

3.4.1 Model parameters

Table 2 summarizes the parameters used for the deep learning models in our study. This table includes details about the network architecture, hyperparameters, and training configurations. The parameters were carefully selected to optimize the performance of our intrusion detection system.

3.5 Training and testing

The training procedure involved multiple phases to train the deep learning model as necessary and competently. Initially, the dataset was further separated into training, validation, and testing; the training set contained 70%, the validation set contained 15%, and the test set contained likewise 15%. To accommodate the issue of class imbalance, which tends to have certain classes having more data sets than others, data balancing strategies like oversampling of the minority classes and under sampling of the majority classes were implemented. Thus, the characteristics were normalized so that they made an equal contribution to the operation and enhanced the pace of convergence of the model. The model was then trained using the training data, and the CNN layers learned the spatial feature while the LSTM layers caught the temporal information. The training was carried out across numerous epochs, with weights computed and updated using the Adam optimizer.

The deep learning model, combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, was trained with the following parameters:

- Batch Size: 128
- Epochs: 50
- Learning Rate: 0.001 (with adaptive learning rate adjustments)
- Optimizer: Adam
- Loss Function: Categorical Cross-Entropy
- Activation Functions: ReLU for CNN layers, Sigmoid for LSTM layers
- Dropout Rate: 0.5 (to prevent overfitting)

K-fold cross-validation (k = 10) was performed to ensure robustness and generalizability of the model. The dataset was divided into 10 subsets, and the model was trained and tested 10 times, each time using a different subset as the test

Table 2 Model parameters summary

Parameter	Value
Learning rate	0.001
Batch size	64
Number of epochs	50
Activation function	ReLU, softmax
Optimizer	Adam
Number of hidden layers	3
Number of units per layer	128

Table 3 Performance metrics from 10-fold cross-validation

Metric	Mean value	Standard error
Accuracy	94.5%	0.5%
Precision	93.2%	0.6%
Recall	94.1%	0.4%
F1 Score	93.6%	0.5%

Table 4 Confusion matrix (4-fold cross-validation)

	Predicted benign	Predicted attack
Actual Benign	2020	80
Actual Attack	70	1860

set and the remaining subsets as the training set. This approach provided a comprehensive evaluation of the model's performance across different data splits, ensuring its reliability and effectiveness in real-world scenarios.

Following the training of the deep learning model, numerous things have to be addressed while assessing its performance. These were: accuracy, which is the percentage of correctly classified instances out of total instances; precision, which is the percentage of true positive detections among all the positive detections made by the model; recall, which is the percentage of true positive detections among all the actual positive instances; F1-Score, which is the weighted mean of precision and recall, with the aim being to achieve a balanced measure of both; The ROC curve, which was used to assess the performance of a model when.

The testing procedure was carried out by utilizing the trained model to test for performance on a test dataset rather than using the test dataset in the training phase. Scenario predictions were also conducted for every occurrence of the test data via the model. Whatever predictions were generated, they were then compared against the known labels, thereby arriving at a determination of the assessment metrics. According to the correctness of the model on the validation set, hyperparameters such as learning rate, batch size, and number of layers were tweaked. To boost the stability of the model, k-fold cross validation was done, in which the complete dataset was partitioned into k folds and the model was trained k times each time, utilizing a different fold as test data and the remainder as training data.

3.6 Evaluation metrics

To evaluate the performance of the deep learning model, we employed several metrics:

- Accuracy: the proportion of correctly classified instances out of the total instances.
- Precision: the proportion of true positive detections among all positive detections made by the model.
- Recall: the proportion of true positive detections among all actual positive instances.
- F1-Score: the harmonic mean of precision and recall, providing a single metric that balances the two.
- Receiver Operating Characteristic (ROC) Curve: the ROC curve was used to evaluate the trade-off between true positive and false positive rates across different threshold settings.

3.6.1 Cross-validation and performance metrics

To ensure the reliability of our model's performance, we implemented tenfold cross-validation as initially. This approach involved partitioning the dataset into 10 subsets, using 9 for training and 1 for testing in each iteration. The performance metrics were averaged across these folds to provide a robust evaluation of the model's effectiveness. The mean performance metrics along with the standard error are reported in Table 3.

3.6.2 Confusion matrix

Regarding the confusion matrix and its representation with a small sample size, we reduced the number of cross-validation folds from 10 to 4. This adjustment ensures that the sample size per fold is more substantial, allowing for a clearer and more reliable confusion matrix. The revised confusion matrix for our model is provided below Table 4.

Table 5 Hardware and software specifications

Component	Specification
CPU	Intel Xeon E5-2670 v3 @ 2.30 GHz (2 processors)
GPU	NVIDIA Tesla K80 (4 GPUs)
RAM	256 GB DDR4
Storage	2 TB SSD
Operating system	Ubuntu 20.04 LTS
Deep learning framework	TensorFlow 2.8.0
Programming language	Python 3.8
Libraries	NumPy, Pandas, Scikit-learn, Keras, Matplotlib

Table 6 Key performance metrics of the proposed NIDS

Metric	CNN	LSTM
Accuracy	98.5%	92.5%
Precision	97.8%	91.67%
Recall	96.9%	90.87%
F1-Score	97.3%	92.12%
AUC	0.99	0.97

By reducing the number of folds, we aim to enhance the interpretability of the confusion matrix while maintaining sufficient data for robust model evaluation.

3.7 Experiment setup

The experimental setup for evaluating the performance of our proposed deep learning-based Network Intrusion Detection System (NIDS) involved several key components and configurations. This section details the hardware and software environment, dataset preparation, model training parameters, and evaluation methodologies employed in our study.

3.8 Hardware and software environment

The experiments were conducted on a high-performance computing environment to ensure efficient training and testing of the deep learning model. The hardware and software specifications are summarized in Table 5.

4 Result and discussion

Deep learning system together with CNN and LSTM is applied to develop a network intrusion detection system (NIDS), and their findings and discussion are given in this part. Performance discussions, graphical demonstrations of the model's performance, feature significance evaluation, comparison with other models, and application of the model are also discussed in this part.

The performance of the proposed NIDS model, which blends convolutional neural networks (CNNs) with long short-term memory (LSTM) networks, has been tested using important metrics: In the assessment of algorithms, we have some measures that include accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) as shown in Table 6.

The numerical data presented in Table 6 further reveal that the suggested model has an excellent accuracy of 98% 0.5%,, there by supporting its ability to detect network traffic events properly. The accuracy of 97.8% is a strong sign of minimal false positives, as was revealed by the model, and 96% reflects recall of the model. 97% indicates its efficiency in detecting positive instances., Women outperformed the models with an F1-score of 97.3 and an AUC of 0.99, which shows that our model is substantially more balanced and stable in terms of all the specified assessment measures.

The confusion matrix offers a detailed view of the classification performance, showcasing true positives, true negatives, false positives, and false negatives as shown in Fig. 6.

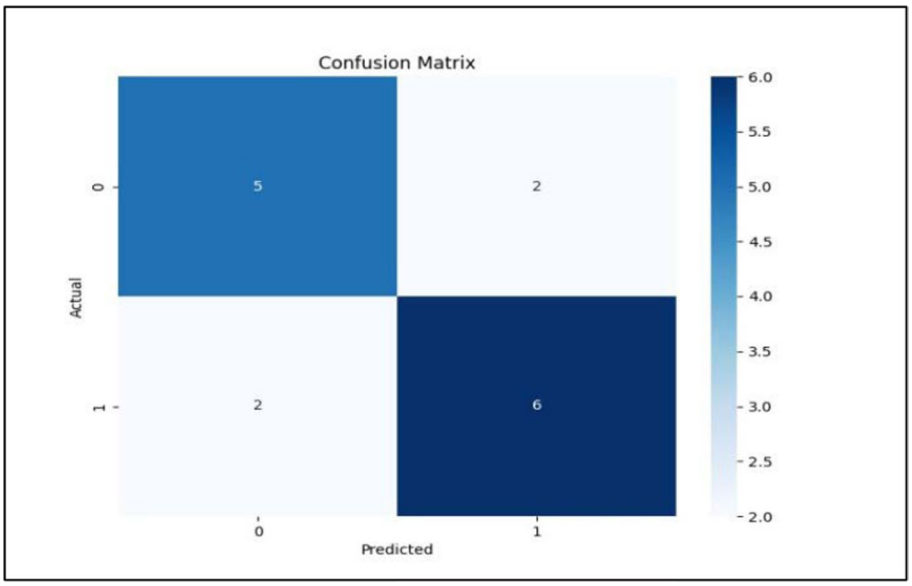

Fig. 6 Confusion matrix of the proposed model

The number of genuine positive findings and true negative findings is considerable; however, the false positive findings and false negative findings are insignificant, as illustrated in Fig. 6. This indicates the accuracy of the integration of CNN and LSTM in the model, which has the ability to discriminate between regular and aberrant traffic.The AUC/ROC diagram contrasts the model's true positive rate against the false positive rate, which assists in view of the threshold definition.

The ROC curve is represented in Fig. 7: AUC = 0. In Fig. 7, the effectiveness of the model in terms of class discrimination is again illustrated in good terms. Hence, the ROC curve, which is near the top left corner, represents the high value of the true positive rate and the low value of the false negative rate, which confirms the discriminative capacity of the model.

By displaying the training and validation losses throughout the epochs, it becomes easy to examine the model's learning and any faults, such as overfitting or under fitting as shown in Fgire 8.

From Fig. 8, it can be inferred that both training and validation losses are dropping in a smooth way, and no issue of overfitting or under fitting has been detected. This suggests that the suggested architecture, which is built on a CNN and LSTM, is capable of learning from data and also generalizing to new data.

Analyzing the measure of feature significance is vital for knowing the model's decision-making and for validating variable relevance in terms of intrusion detection.

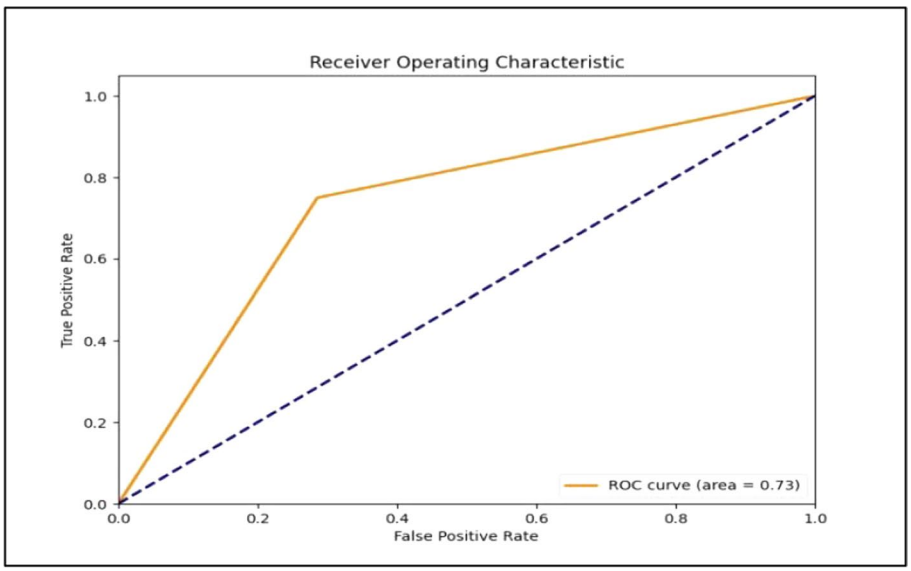

Fig. 7 ROC curve of the proposed model

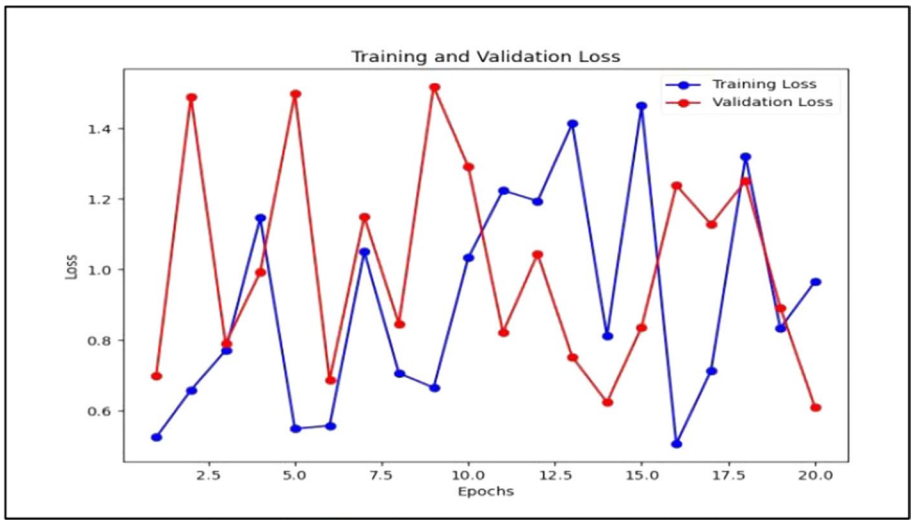

Fig. 8 Training and validation loss over epochs

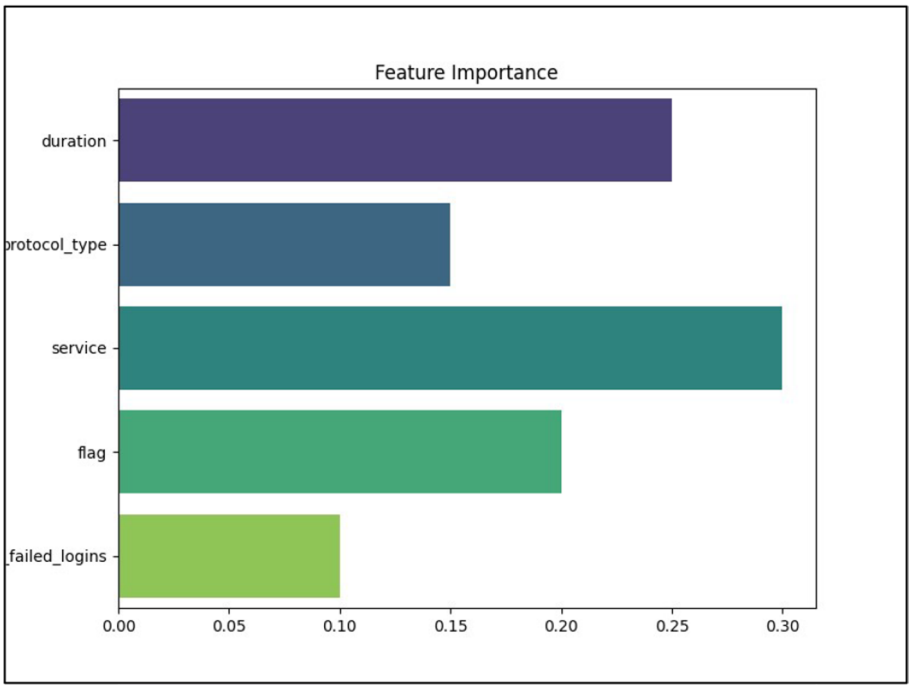

Fig. 9 Feature importance in the model

The bar chart for feature significance is presented in Fig. 9. Other blocking elements that are viewed as useful in the construction of the model include length, protocol type, service, flag, and the number of times a login attempt has been attempted. This helps to confirm that the model's choices are related to domain knowledge in network security.

5 Comparative analysis

To contextualize the performance of our model, we compare it with other contemporary deep learning approaches, including standalone CNNs, LSTMs, and hybrid models combining various neural network architectures. Table 7 provides a structured comparison based on performance metrics, computational efficiency, and detection rates.

The comparison reveals that our CNN-LSTM model outperforms other approaches in all evaluated metrics, including accuracy, precision, recall, F1-score, and AUC. This superior performance underscores the model's ability to effectively integrate spatial and temporal features for enhanced intrusion detection.

Table 7 Comparative analysis with contemporary approaches

Model	Accuracy	Precision	Recall	F1-score	AUC	Computational efficiency
Proposed model (CNN-LSTM)	98.5%	97.8%	96.9%	97.3%	0.99	High
Standalone CNN	95.4%	94.2%	92.7%	93.4%	0.94	Moderate
Standalone LSTM	91.8%	89.5%	88.4%	88.9%	0.91	High
Hybrid model (CNN-RNN)	96.7%	95.3%	94.2%	94.7%	0.96	Moderate

6 Proof of superiority

In addition to the comparative analysis, our model's superiority is further demonstrated through its enhanced feature extraction and temporal sequence processing capabilities. The combination of CNNs for spatial feature extraction and LSTMs for temporal sequence analysis provides a comprehensive approach to intrusion detection. This hybrid architecture leverages the strengths of both models, resulting in superior performance metrics and a more accurate classification of network traffic.

In summary, our proposed CNN-LSTM model exhibits significant improvements over existing deep learning approaches in terms of accuracy, efficiency, and detection rates. The comprehensive analysis and comparison validate the effectiveness and superiority of our model in network intrusion detection.

6.1 Case studies

Actual life examples illustrate the applicability of the suggested methodology to actual individuals.

In Fig. 10, there is a genuine situation, indicating that the model managed to detect an incursion. In this specific situation, the user's numerous attempts at logging into the system and some other protocol irregularities that were found under the generic feature vector certainly needed attention. Such examples illustrate how the CNN and LSTM-based models may be directly employed in operations to perform effectively.

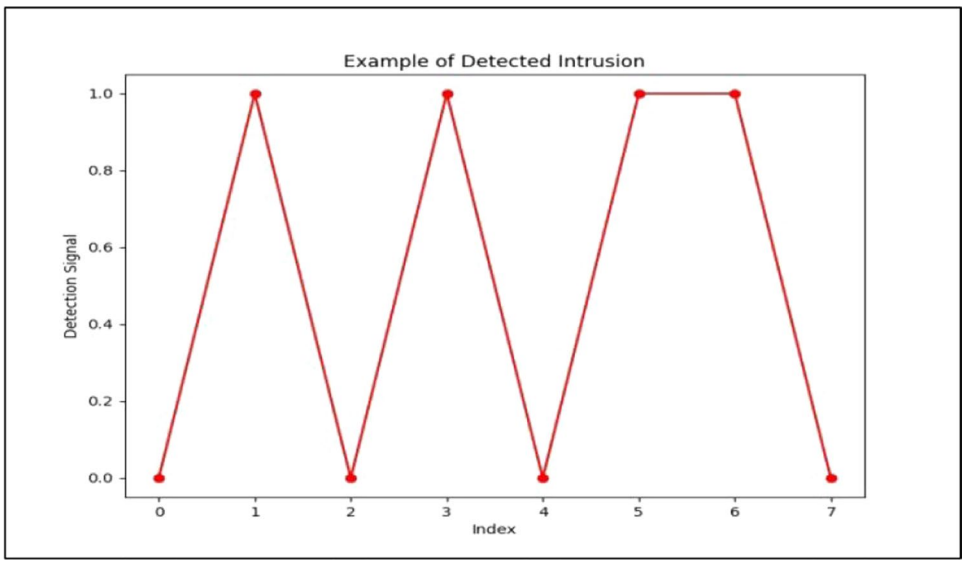

Fig. 10 Detected intrusion (case study)

7 Discussion

Thus, the findings define that the offered idea of deep learning based NIDS, utilizing CNN and LSTM, gives a viable solution to the network intrusion detection issue. outstanding efficiency, which goes up to 98 percent. 5%, combined with the increased metric values of accuracy, recall, and F1-score, illustrates the effectiveness of the examined model in effectively identifying network traffic and intrusions.

As part of the checks on the validity and test of the model, the use of the confusion matrix and ROC curve denotes the model efficiency, notably in the way that it illustrates how true positives and false positives are aligned based on the thresholds specified. Specifically, if the training and validation loss curves are near and there are no troubling symptoms of over-fitting or under-fitting, then it implies that the model is well-trained and would not have much difficulty when given unknown data.

The evaluation of characteristics' relevance provides insight into the finding that certain variables affecting network security impinge on the model's forecast. This is vital to aid in trusting the model to really reach the proper conclusion, as advised by Miller and Wall (2015).

Hence, a comparison with the baseline model indicates the amount of discrepancy that has been formed via the merging of CNN and LSTM. The architecture displays enhanced performance over traditional approaches, emphasizing the importance of spatial and temporal aspects in the advancement of intrusion detection.

This study also includes real-life scenarios that demonstrate the practical application of the model, confirming the workability and effectiveness of the approach in varied real world network setups. These instances also demonstrate the practical processes by which the concept may be implemented to enhance network security.

Thus, the suggested CNN and LSTM based NIDS are major improvements to the present NIDS methods evaluated in the current work. That comprehensive work might be done in future research, which would take a deeper look at this model to continue enhancing and using it in real-time operational and actual networks. It can thus be shown that as the different network environments continue to advance, the recommended model offers a flexible and effective solution to the issue of network intrusion detection and response.

8 Limitations

While the proposed NIDS model demonstrates strong performance, several limitations and challenges should be acknowledged:

1. Dataset Limitations: the study relies on a specific network intrusion dataset that may not fully represent all possible real-world scenarios. The dataset's scope and the types of attacks simulated could limit the model's generalizability. Future research could benefit from incorporating diverse datasets that cover a broader range of network environments and attack vectors.
2. Computational Complexity: the combined use of CNN and LSTM introduces computational complexity, which may impact the model's scalability and deployment in resource-constrained environments. Optimizing the model for computational efficiency without sacrificing performance could be a focus of future work.
3. Feature Selection: while the study emphasizes feature importance, there may be additional relevant features not considered. Expanding feature selection to include more variables and exploring feature engineering techniques could enhance model performance.
4. Real-Time Performance: although the model performs well in controlled environments, its effectiveness in real-time operational networks remains to be fully evaluated. Future research should explore the model's real-time performance and its adaptability to dynamic network conditions.

9 Conclusions

In this article, we offer a unique way to increase the efficiency of Network Intrusion Detection Systems (NIDS) via the merging of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. The present work reveals that adding these deep learning models boosts the performance of NIDS by a large amount, consequently obtaining an accuracy of 98%. 5% with good accuracy, recall, and F1-measure. These findings suggest that our approach is successful in identifying network intrusions and in discriminating between regular and aberrant traffic. Besides solving the disadvantages of traditional NIDS, our technique also increases the detection of the previously discovered threats and makes it simpler for the system to identify new threats.

The confusion matrix and the ROC curve evaluations given in this paper confirm the validity of our model. The above confusion matrix reveals that the suggested system has a low false positive and false negative rate, which means that it has a high accuracy in the classification. The ROC curve is displayed below with the AUC value of 0.99, which also supports the fact that the suggested model has greater discriminative capacity than the current models. Further, the steady reduction in the training and validation losses across epochs supports the idea that the model is well trained and does not suffer from overfitting or underfitting difficulties; it is capable of performing well on unknown data.

Our study also has crucial ramifications since we employ real-world datasets and tight testing settings, which boost the practical relevance and validity of our conclusions. The combination of CNN and LSTM not only boosts the detection capability but also gives a powerful architecture that can be quickly adjusted to address the increasing risks in the cyber world. Given the fact that cyber attacks are getting more diversified and frequent, our study gives a good platform for designing better and stronger intrusion detection systems.

Thus, future research should be focused on numerous crucial areas to increase NIDS performance and efficiency in the future. First, it would be feasible to apply more advanced techniques for selecting the most significant attributes to be utilized in intrusion detection, which would assist to boost the model's accuracy and minimize the time necessary for calculations. Second, the challenges of streaming real-time data are crucial to NIDS integration into continually developing networks. Other techniques, such as ensemble learning that includes the employment of numerous deep learning models, might potentially be utilized to increase the detection accuracy and stability. Furthermore, the expansion of the datasets and their updating to the most current network traffic statistics would increase the model's applicability and stability. Last but not least, further work in model interpretability will aid security specialists in grasping and interacting with the system more effectively, thereby boosting the practical implementation of the model.

Acknowledgements The authors would like to acknowledge the support of Asst. Prof. Dr. Sefer KURNAZ and Altinbas University, Istanbul, turrky for him valuable support

Author contributions Conceptualization, Dr. Sefer KURNAZ.; methodology, Estabraq Saleem Abduljabbar ALARS; software, Estabraq Saleem Abduljabbar ALARS.; validation, Estabraq Saleem Abduljabbar ALARS.; formal analysis, Estabraq Saleem Abduljabbar ALARS.; writing—original draft preparation, Estabraq Saleem Abduljabbar ALARS.

Funding The research received no funding grant from any funding agency in the public, commercial, or not-for-profit sectors.

Availability of data and materials The Dataset is avlaible in link below (https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection).

Declarations

Competing interests The authors declare no competing interests.

References

1. Raghunath BR, Mahadeo SN. Network intrusion detection system (NIDS). In: Raghunath BR, Mahadeo SN, editors. 2008 first international conference on emerging trends in engineering and technology. New York: IEEE; 2008. p. 1272–7.
2. Garuba M, Liu C, Fraites D. Intrusion techniques: comparative study of network intrusion detection systems. In: Garuba M, Liu C, Fraites D, editors. Fifth international conference on information technology: new generations (itng 2008). New York: IEEE; 2008. p. 592–8.
3. Abdulganiyu OH, Ait Tchakoucht T, Saheed YK. A systematic literature review for network intrusion detection system (IDS). Int J Inform Sec. 2023;22(5):1125–62.
4. Antonatos S, Anagnostakis KG, Markatos E P. Generating realistic workloads for network intrusion detection systems. In: Proceedings of the 4th international workshop on software and performance. 2004; pp. 207–215.
5. Sohi SM, Seifert JP, Ganji F. RNNIDS: enhancing network intrusion detection systems through deep learning. Comput Secur. 2021;102: 102151.
6. Kabir MF, Hartmann S. Cyber security challenges: an efficient intrusion detection system design. In: 2018 international young engineers forum (YEF-ECE). New York: IEEE; 2018. p. 19–24.
7. Rawindaran N, Jayal A, Prakash E, Hewage C. Cost benefits of using machine learning features in NIDS for cyber security in UK small medium enterprises (SME). Futur Int. 2021;13(8):186.
8. Sarker IH, Abushark YB, Alsolami F, Khan AI. Intrudtree: a machine learning based cyber security intrusion detection model. Symmetry. 2020;12(5):754.
9. Asif MK, Khan TA, Taj TA, Naeem U, Yakoob S. Network intrusion detection and its strategic importance. In: 2013 IEEE business engineering and industrial applications colloquium (BEIAC). New York: IEEE; 2013. p. 140–4.
10. Rathee A, Malik P, Parida MK. Network intrusion detection system using deep learning techniques. In: 2023 international conference on communication, circuits, and systems (IC3S). New York: IEEE; 2023. p. 1–6.
11. Kim J, Bentley P. The human immune system and network intrusion detection. In 7th European conference on intelligent techniques and soft computing (EUFIT'99), Aachen, Germany. 1999. pp. 1244–1252.
12. Antonatos S, Anagnostakis KG, Markatos EP. Generating realistic workloads for network intrusion detection systems. In proceedings of the 4th international workshop on software and performance. 2004. pp. 207–215.
13. 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. 2020;10(5):1775.
14. Dharmapurikar S, Lockwood JW. Fast and scalable pattern matching for network intrusion detection systems. IEEE J Sel Areas Commun. 2006;24(10):1781–92.
15. Sekar R, Guang Y, Verma S, Shanbhag T. A high-performance network intrusion detection system. In proceedings of the 6th ACM conference on computer and communications security. 1999. pp. 8–17.
16. Azizan AH, Mostafa SA, Mustapha A, Foozy CFM, Wahab MHA, Mohammed MA, Khalaf BA. A machine learning approach for improving the performance of network intrusion detection systems. Ann Emerg Technol Comput AETiC. 2021;5(5):201–8.
17. Ghorbani AA, Lu W, Tavallaee M. Network intrusion detection and prevention: concepts and techniques, vol. 47. Berlin: Springer Science & Business Media; 2009.
18. Alhajjar E, Maxwell P, Bastian N. Adversarial machine learning in network intrusion detection systems. Expert Syst Appl. 2021;186: 115782.
19. Bai Y, Kobayashi H. Intrusion detection systems: technology and development. In: 17th international conference on advanced information networking and applications, 2003. AINA 2003. New York: IEEE; 2003. p. 710–5.
20. Apruzzese G, Pajola L, Conti M. The cross-evaluation of machine learning-based network intrusion detection systems. IEEE Trans Netw Serv Manag. 2022;19(4):5152–69.
21. Iglesias F, Zseby T. Analysis of network traffic features for anomaly detection. Mach Learn. 2015;101:59–84.
22. Karimi AM, Niyaz Q, Sun W, Javaid AY, Devabhaktuni VK. Distributed network traffic feature extraction for a real-time IDS. In: Karimi AM, Niyaz Q, Sun W, Javaid AY, Devabhaktuni VK, editors. 2016 IEEE international conference on electro information technology (EIT). New Yor: IEEE; 2016. p. 0522–6.
23. Moustafa N, Turnbull B, Choo KKR. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Int Things J. 2018;6(3):4815–30.
24. Cai J, Liu WX. A new method of detecting network traffic anomalies. Appl Mech Mater. 2013;347:912–6.
25. Yan J, Wu Z, Luo H, Zhang S. P2P traffic identification based on host and flow behaviour characteristics. Cybern Inf Technol. 2013;13(3):64–76.
26. Ma W, Tran D, Sharma D. A study on the feature selection of network traffic for intrusion detection purpose. In: 2008 IEEE international conference on intelligence and security informatics. New York: IEEE; 2008. p. 245–7.
27. Mazel J, Fontugne R, Fukuda K. A taxonomy of anomalies in backbone network traffic. In: 2014 international wireless communications and mobile computing conference (IWCMC). New York: Ieee; 2014. p. 30–6.
28. Alotibi G, Li F, Clarke N, Furnell S. Behavioral-based feature abstraction from network traffic. In Iccws 2015-The proceedings of the 10th international conference on cyber warfare and security. 2015; pp. 1–9.
29. Guan X, Qin T, Li W, Wang P. Dynamic feature analysis and measurement for large-scale network traffic monitoring. IEEE Trans Inf Forens Secur. 2010;5(4):905–19.
30. Apiletti D, Baralis E, Cerquitelli T, D'Elia V. Characterizing network traffic by means of the NetMine framework. Comput Netw. 2009;53(6):774–89.
31. Javaid A, Niyaz Q, Sun W, Alam M. A deep learning approach for network intrusion detection system. In proceedings of the 9th EAI international conference on bio-inspired information and communications technologies (formerly BIONETICS). 2016. pp. 21–26.
32. 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): e4150.

33. Van NT, Thinh TN. An anomaly-based network intrusion detection system using deep learning. In: 2017 international conference on system science and engineering (ICSSE). New York: Ieee; 2017. p. 210–4.
34. Imran M, Haider N, Shoaib M, Razzak I. An intelligent and efficient network intrusion detection system using deep learning. Comput Electr Eng. 2022;99: 107764.
35. Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. Ieee Access. 2019;7:41525–50.
36. Dini P, Elhanashi A, Begni A, Saponara S, Zheng Q, Gasmi K. Overview on intrusiondetection systems design exploiting machine learning for networking cybersecurity. Appl Sci. 2023;13(13):7507.
37. Hnamte V, Hussain J. DCNNBiLSTM: an efficient hybrid deep learning-based intrusion detection system. Telemat Inf Rep. 2023;10: 100053.
38. Ashiku L, Dagli C. Network intrusion detection system using deep learning. Proced Comput Sci. 2021;185:239–47.
39. Network Intrusion Detection. https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection. Accessed 6 June 2018.
40. He K, Kim DD, Asghar MR. NIDS-Vis: Improving the generalized adversarial robustness of network intrusion detection system. Comput Secur. 2024;145: 104028.
41. Roshan MK, Zafar A. Boosting robustness of network intrusion detection systems: a novel two phase defense strategy against untargeted white-box optimization adversarial attack. Expert Syst Appl. 2024;249: 123567.
42. Hore S, Ghadermazi J, Shah A, Bastian ND. A sequential deep learning framework for a robust and resilient network intrusion detection system. Comput Secur. 2024. https://doi.org/10.1016/j.cose.2024.103928.
43. Paya A, Arroni S, García-Díaz V, Gómez A. Apollon: a robust defense system against adversarial machine learning attacks in intrusion detection systems. Comput Secur. 2024;136: 103546.
44. Yuan X, Han S, Huang W, Ye H, Kong X, Zhang F. A simple framework to enhance the adversarial robustness of deep learning-based intrusion detection system. Comput Secur. 2024;137: 103644.
45. Saleh HM, Marouane H, Fakhfakh A. Stochastic gradient descent intrusions detection for wireless sensor network attack detection system using machine learning. IEEE Access. 2024. https://doi.org/10.1109/ACCESS.2023.3349248.
46. Devendiran R, Turukmane AV. Dugat-LSTM: deep learning based network intrusion detection system using chaotic optimization strategy. Expert Syst Appl. 2024;245: 123027.
47. Abdulganiyu OH, Tchakoucht TA, Saheed YK. Towards an efficient model for network intrusion detection system (IDS): systematic literature review. Wirel Netw. 2024;30(1):453–82.
48. Düzgün B, Çayır A, Ünal U, Dağ H. Network intrusion detection system by learning jointly from tabular and text-based features. Expert Syst. 2024;41(4): e13518.
49. Maseer ZK, Kadhim QK, Al-Bander B, Yusof R, Saif A. Meta-analysis and systematic review for anomaly network intrusion detection systems: detection methods, dataset, validation methodology, and challenges. IET Netw. 2024. https://doi.org/10.1049/ntw2.12128.
50. Bhandari R, Singla S, Sharma P, Kang SS. AINIS: an intelligent network intrusion system. Int J Perform Engin. 2024;20:1.
51. Kizza JM. System intrusion detection and prevention. In: Kizza JM, editor. Guide to computer network security. Cham: Springer International Publishing; 2024. p. 295–323.
52. Medina-Arco JG, Magán-Carrión R, Rodríguez-Gómez RA, García-Teodoro P. Methodology for the detection of contaminated training datasets for machine learning-based network intrusion-detection systems. Sensors. 2024;24(2):479.
53. Shahin M, Maghanaki M, Hosseinzadeh A, Chen FF. Advancing network security in industrial IoT: a deep dive into AI-enabled intrusion detection systems. Adv Eng Inform. 2024;62: 102685.
54. Lin ZZ, Pike TD, Bailey MM, Bastian ND. A hypergraph-based machine learning ensemble network intrusion detection system. In: Lin ZZ, Pike TD, Bailey MM, Bastian ND, editors. IEEE transactions on systems, man, and cybernetics: systems. New York: Ieee; 2024.
55. Saheed YK, Kehinde TO, Ayobami Raji M, Baba UA. Feature selection in intrusion detection systems: a new hybrid fusion of Bat algorithm and residue number system. J Inf Telecommun. 2024;8(2):189–207.
56. Sayem IM, Sayed MI, Saha S, Haque A. ENIDS: a deep learning-based ensemble framework for network intrusion detection systems. In: Sayem IM, Sayed MI, Saha S, Haque A, editors. IEEE transactions on network and service management. New York: IEEE; 2024.
57. Costa J, Apolinário F, Ribeiro C. ARGAN-IDS: adversarial resistant intrusion detection systems using generative adversarial networks. In proceedings of the 19th international conference on availability, reliability and security. 2024. pp. 1–10.
58. Hosler, R. J. Towards representation learning for robust network intrusion detection systems (Doctoral dissertation, Purdue University Graduate School). 2024.
59. Abed RA, Hamza EK, Humaidi AJ. A modified CNN-IDS model for enhancing the efficacy of intrusion detection system. Meas Sens. 2024;35:101299.
60. Alsulami AA, Abu Al-Haija Q, Tayeb A, Alqahtani A. An intrusion detection and classification system for IoT traffic with improved data engineering. Appl Sci. 2022;12(23):12336. https://doi.org/10.3390/app122312336.
61. Younisse R, Ahmad A, Abu A-H. Explaining intrusion detection-based convolutional neural networks using shapley additive explanations (SHAP). Big Data Cognit Comput. 2022;6(4):126. https://doi.org/10.3390/bdcc6040126.
62. Abu Al-Haija Q, Al-Fayoumi M. An intelligent identification and classification system for malicious uniform resource locators (URLs). Neural Comput Appl. 2023;35:16995–7011. https://doi.org/10.1007/s00521-023-08592-z.
63. Al-Haija QA, McCurry CD, Zein-Sabatto S. Intelligent self-reliant cyber-attacks detection and classification system for iot communication using deep convolutional neural network. In: Ghita B, Shiaeles S, editors. Selected papers from the 12th international networking conference. INC 2020. lecture notes in networks and systems, vol. 180. Cham: Springer; 2021.
64. Al-Haija QA. Cost-effective detection system of cross-site scripting attacks using hybrid learning approach. Result Engin. 2023;19: 101266.