Egyptian Informatics Journal 25 (2024) 100448

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Cyber vulnerabilities detection system in logistics-based IoT data exchange

Ahmed Alzahrani [a], Muhammad Zubair Asghar [b,*]

[a] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[b] Gomal Research Institute of Computing, Faculty of Computing, Gomal University, D.I.Khan, Pakistan

ARTICLE INFO

Keywords:
Hybrid deep learning
Feature selection
IoT-based vulnerabilities
Logistics

ABSTRACT

Modern-day digitalization has a profound impact on business and society, revolutionizing logistics. Supply chain digitalization improves transparency, speed, and cost-effectiveness, increasing tech adoption—transportation benefits from IoT-driven shipment tracking and web data storage. However, cyber threats target IoT data by exploiting cyber vulnerabilities. Although ML/DL approaches have showed potential in finding IoT vulnerabilities, the difficulty of selecting appropriate features remains. Existing research has produced surprising outcomes, and deep neural networks have been utilised to extract characteristics without taking sequence information into account. To address this, the paper presents a unique approach for accurate IoT vulnerability identification that combines deep learning and better feature selection. On the BoT-IoT dataset, the LSTM + CNN model achieved 95.73 % accuracy. This approach has the ability to successfully anticipate IoT based vulnerabilities by leveraging benchmark data, selecting relevant features, and enhancing overall system performance.

1. Introduction

The interconnected network of human, mechanical, resource, and technology nodes involved in producing and distributing goods is referred to as the "supply chain" (SC). This includes everything from raw supplies to ultimate delivery. Effective SC management is essential for long-term resource supply [1]. Decision-makers organise a variety of duties related to material acquisition, transportation, and timely delivery. Logistics is defined by the Council of Supply Chain Management Professionals as a process inside the supply chain that assures the effective movement and storage of goods, services, and information to meet consumer needs [2]. Natural calamities and the COVID-19 epidemic impeded product flow, needing problem-solving expertise. During the COVID-19 pandemic, the growing logistics sector becomes vulnerable to cyber attacks, owing to its reliance on online storage. For exmaple, Intel 471, a well-known source of cybercrime intelligence, has issued a warning about cybersecurity threats in the supply chain. They highlight network access brokers who are reportedly selling credentials obtained through remote access vulnerabilities obtained from shipping businesses. The repercussions could be detrimental to the global consumer economy [3].

1.1. Research motivation (need of cyber vulnerabilities detection system in logistics-based IoT data exchange)

The Internet of Things (IoT) has become an essential component of the logistics business, with sensors and actuators used to track and monitor shipments, optimise transportation routes, and improve warehouse management. However, as IoT devices become increasingly connected, the logistics industry has become more exposed to hacks [4]. Attackers might exploit cyber vulnerabilities in logistics-based IoT data exchange to obtain unauthorised access to sensitive data, disrupt operations, or even inflict physical damage. For example, in 2017, hackers targeted the Maersk shipping corporation, causing significant disruption to its global operations [5]. Recognizing the existence of cyber risks is essential for ensuring cybersecurity within the logistics sector. Difficulties may arise as a result of working with external partners over uncontrolled networks during the transportation process. The Internet of Things (IoT) devices are driving the change of the logistics business. IoT is a network of heterogeneous components that enables intelligent systems and services to recognize, collect, disseminate, and interpret information. The term "things" in the Internet of Things (IoT) refers to electronic gadgets that may gather and send data, such as sensors, wearables, intelligent appliances, carbon monoxide alarms, radio frequency identification (RFID), cardiac trackers, gyroscopes, cellphones, and many others. Every day, there are more and more devices added to

* Corresponding author.
E-mail addresses: aalzahrani9@kau.edu.sa (A. Alzahrani), mzubairgu@gmail.com, mzubair@gu.edu.pk (M.Z. Asghar).

https://doi.org/10.1016/j.eij.2024.100448
Received 2 September 2023; Received in revised form 2 December 2023; Accepted 20 January 2024
Available online 1 February 2024

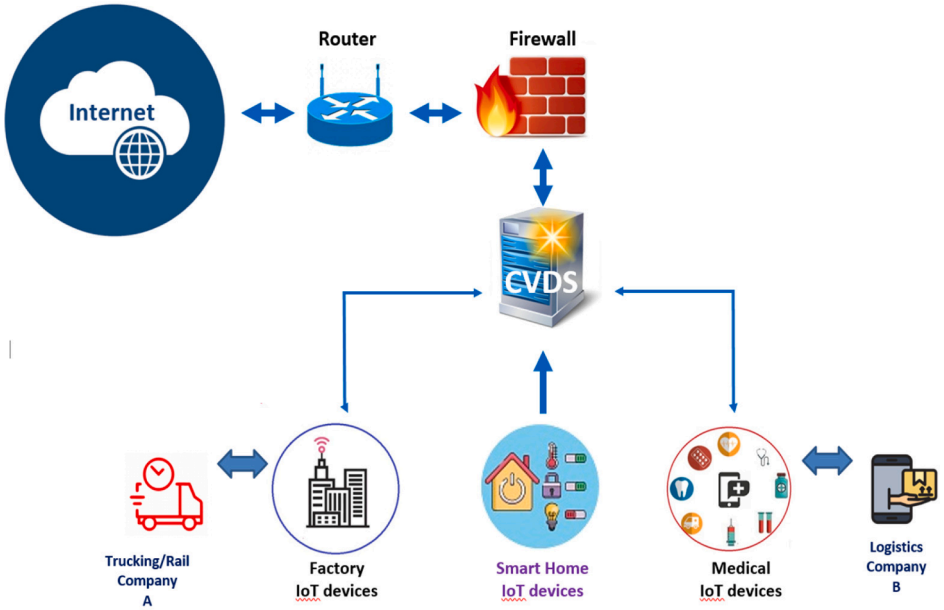

Fig. 1. Logistics-based IoT Network Architecture.

IoT networks. It is anticipated that the number of Internet of Things (IoT) devices in use across the globe would almost triple from 9.7 billion in the year 2020 to more than 29 billion in the year 2030 [6]. In many ways, On our everyday activities the IoT has a powerful effect, including socioeconomic, business, and financial ones. Revenue from the Internet of Things (IoT) is expected to expand from $892.2 billion in 2018 to $4 trillion in 2025, a significant contributor to the expansion of the modern economy [7].

Cybersecurity protects computers, servers, networks, mobile devices, and data from cyber threats that could result in data breaches. To safeguard supply chain assets during logistics-based IoT data exchange from unauthorised access, disclosure, disruption, and data manipulation, effective security measures including various technologies, practises, and protocols are required. Furthermore, considerable improvements to existing data and communication device security concepts are also necessary [8]. Current security solutions like encrypted data, authentication, permissions, vulnerability scanning, and content filtering are time-consuming and inappropriate for a large system with several interconnected systems, each of which presents its own threat. The Mirai botnet, for example, is a rare form of malware that exploits logistics-based Internet of Things (IoT) devices to launch massive distributed denial-of-service (DDoS) attacks [9]. A growing number of IP cameras have been infected with a variation of the Mirai code called Persirai thingbot [10]. Therefore, the logistics sector has to prioritize cybersecurity measures in an era of increasingly devastating cyber attacks.

Fig. 1 depicts Logistics-based IoT network architecture [3]. Cyber Vulnerabilities Detection System (CVDS) are critical in monitoring hosts or networks for security breaches and notifying administrators as soon as they are found. However, stealth system development is still in its early phases, and various hurdles must be overcome in order to attain high accuracy and low false alert rates. CVDS are grouped into three types based on their detection methods: Signature, Anomaly, and Specification. An alarm is triggered by Signature-I CVDS when it detects a match between network traffic patterns and previously stored attack patterns. It is highly accurate and has a low false alert rate, but it has difficulty detecting new forms of threats. Specification-based IDS, on the other hand, detects malicious activity by comparing network traffic behavior to the current rule set and values.

At the moment, these requirements are determined manually by security specialists [2–7]. However, logistics-based IoT devices generate a large amount of data, which may overwhelm traditional data collection, processing, and storage techniques [11,12] The logistics-based IoT's data heterogeneity offers issues for conventional data processing technologies. To properly predict and assess huge amounts of data, new strategies to deal with this overwhelming volume of information must be devised. Furthermore, When put up against the anticipated threats to usage protection, the numerous attacks undertaken by adversaries to circumvent the default setup suddenly prove to be ineffective [11]. Not only do existing security components need to be upgraded to meet the logistics-based IoT environment [12], furthermore, it is essential in logistics-based IoT context to enhance security of network and prepare embedded cognition in a new unique way.

To train proactively from data and anticipate future occurrences, deep learning (DL) technique employs a comprehensive set of feature embedding algorithms [11,12]. Over the years, it has been put to use in many different contexts, such as stock market forecasting [14], developing predictive models [1] and classifying DDos attacks [12], and several others [3]. To help network administrators more reliably identify vulnerabilities in logistics-based IoT networks, data scientists work hard to develop practical solutions [13]. Because of this, it is crucial to research and put into practise advanced DL composite models leveraging baseline data accurately for logistics-based IoT vulnerability detection.

1.2. *The goal of the research*

There are a plethora of frameworks and approaches [1,4,5] available for preventing logistics-based IoT cyberattacks. In order to monitor a huge network's logs, which can exceed a billion per day, it is helpful to employ machine learning and deep learning techniques, including supervised and unsupervised. Several researchers [1,4,5,15] have investigated the use of computational approaches, such as machine learning (ML) and other techniques, to analyse logistics-based IoT data and predict potential threats using previously collected information. In this research, we aimed to determine how logistics-based IoT frameworks can classify various threats. This work therefore aimed to investigate how to best implement cyber vulnerabilities detection system in logistics-based IoT data exchange using historical data.

Therefore, the LSTM + CNN composite DL model described in this paper used improved feature selection techniques. A chi-squared (x2) test was used in the initial step to identify suitable determinants for

forecasts of cyber vulnerabilities in logistics-based IoT exchange.

The LSTM model was then implemented since it is excellent at capturing long-term interrelationships within the data environment. Additionally, we enhanced the predictability of logistics-based IoT vulnerabilities by extracting the observed high-rated features using a CNN.

1.3. Problem statements

The authors of [11,15] created an ML method for predicting cyber vulnerabilities in logistics-based IoT data exchange using benchmark data. Specifically, it used a type of ML called as a ensemble ML algorithm to try to foresee security flaws in Internet of Things devices. Selecting effective predictors before employing DL to large data sets, nevertheless, may produce fruitful outcomes. For this reason, it is possible that traditional DL classifiers cannot serve as a reliable method for forecasting cyber vulnerabilities in logistics-based IoT data exchange from benchmark data alone.

For a variety of reasons, such as poor predictive selection of variable the utilization of conventional attributes collection integrated with the techniques of machine learning, it is difficult to use benchmark data to predict future logistics-based IoT risks [13,16]. In addition, when applied to cyber vulnerabilities in logistics-based IoT data exchange forecasting, DL models tend to be less efficient than they might otherwise be due to incorrect predictive variable choice and the absence of composite models.

To address these issues, we examine forecasting cyber vulnerabilities in logistics-based IoT data exchange using historical data as a multi-label classification problem, in which the vulnerabilities are anticipated from a particular set of data. In order to make an accurate prediction regarding the cyber vulnerability, a composite neural network had a stream of training data. Through the use of a composite deep neural network model with carefully selected features, we intend to create an automated method that trains from supplied training data to predict cyber vulnerabilities in logistics-based IoT data exchange efficiently. To do this, we use a hybrid neural network that is fed a training dataset designated as 'D = [d1, d2, d3,… dn].' The goal of this composite neural network is to forecast cyber vulnerabilities in logistics-based IoT data labelled T1 (normal), T2 (information collecting), T3 (denial of service), and T4 (information theft). The goal is to create an automated approach that effectively anticipates IoT-based risks by leveraging the available training data. To forecast cyber vulnerabilities in logistics-based IoT data exchange this method uses a composite deep neural network model integrating optimized/perfect selection of feature.

1.4. Our proposal

In this paper, we present an enhanced composite DL model (LSTM + CNN) that incorporates a feature selection strategy to solve the constraints of the baseline study [11]. Application areas where the deep learning models have demonstrated promise include intrusion detection [11], DDoS attack prediction [12], and extremist association recognition [17].In order to execute the feature selection approach, we performed an x2 test to recognize essential features that significantly contribute to predict court case judgments. The LSTM model, which excels at capturing long-term interrelationships within the data environment, was then integrated. Furthermore, we used a CNN to extract the identified high-rated traits, increasing the accuracy of predicting IoT-based vulnerabilities. Our suggested strategy, which combines the optimal feature selection technique with LSTM and CNN layers, enables accurate predictions of cyber risks from logistics-based IoT data. This comprehensive methodology ensures a strong and efficient answer to logistics-based IoT security challenges.

1.5. Research questions

The research issues examined in order to accurately predict1 logistics-based IoT dangers are summarized in Table 1.

Table 1
Research questions.

Research Questions	Motivation
*RQ.1*How accurately does the LSTM + CNN Composite DL model forecast logistics-based IoT vulnerabilities when given a benchmark dataset?	To examine and apply the deep neural network model, LSTM + CNN, to classify logistics-based IoT-based vulnerabilities.
RQ2, How does the proposed LSTM + CNN model compare with other ML and DL methods?	Using ML and DL classifiers to assess the performance of traditional and sophisticated feature representations.
RQ3: How to evaluate the efficacy of the proposed method in predicting vulnerabilities in logistics-based IoT systems in relation to benchmark research?	The suggested deep learning model, LSTM + CNN, is assessed and compared to various benchmark models to determine its effectiveness.

1.6. Research contributions

The following scientific contributions have been covered in the present research: (i)vulnerabilities in logistics-based IoT networks can be predicted with the use of the x2 test, which ranks and selects the best features., (ii) For logistics-based IoT-based vulnerability prediction, a LSTM + CNN model was deployed., (iii) Analysis of the proposed strategy for forecasting IoT network vulnerabilities in contrast to the performance of conventional ML classifiers., (iv)Predicting logistics-based IoT vulnerabilities across multiple decision-class sets, (v) evaluation of the proposed method in relation to existing DL models and existing studies' findings, and,(vi) the proposed model considerably improved the ability to predict IoT-based vulnerabilities.

The following sections make up the remaining information in this study: A overview of the literature is included in Section 2, Section 3 to the methodology of the proposed approach, Section 4 to the results and discussion, and Section 5 to the conclusion and future scope of the suggested strategy.

2. Related works

The prior research on identifying attacks based on various logistics-based IoT-related vulnerabilities is summarised and evaluated in this part.

2.1. Anomaly detection in IOT networks

Unauthorized access attempts, unexpected data transfers, and irregular device behaviour are examples of anomalous patterns. For this purposed, [18] Highlighted the usage of anomaly detection in IoT networks for intrusion detection. Researchers [19] stress the need of using a multi-layered strategy within CVDS systems. To obtain full threat coverage, this entails merging intrusion detection and prevention systems (IDPS), anomaly detection, signature-based analysis, and machine learning techniques. The literature investigates the integration of several security mechanisms to improve accuracy and reduce false positives. Saba et al., [20] presented system of intrusion detection using CNN based on anomaly to improve IoT security. The method detects intrusions and aberrant traffic behaviour in IoT traffic with 92.85 % accuracy on the BoT-IoT datasets. In their work on anomaly detection, [21] created a technique for DBMS-based intrusion detection. By analyzing the DBMS log table with ML approaches, the researchers created a novel intrusion detection method for DBMS. On their test dataset, they achieved an accuracy of 95.72 %. They intend to investigate detection for more database attacks in the future.

2.2. Cyber vulnerabilities detection in logistics-based IoT data exchange

The study conducted by [15] explores security concerns in digital

supply chains as a result of information technologies that improve efficiency while simultaneously increasing security risks. It fills weaknesses in physical security by focusing on cyber-physical systems. The study of traditional and digital supply chains, risks, and inequities results in a cyber-physical security paradigm.The incorporation of Internet of Things (IoT) technologies in the logistics sector has revolutionised supply chain management efficiency and effectiveness. However, the rapid expansion of IoT devices and the associated data interchange creates a slew of cyber vulnerabilities that pose serious challenges to the security and integrity of logistical operations [22].

The proposed framework [23] combines the Internet of Things (IoT) with smart logistics networks to improve logistics efficiency and security. The framework is made up of three major components: a data gathering and transmission system based on IoE, an intelligent data processing and analysis system, and a secure data transfer system. The framework is tested using a simulation, and the findings show that it can improve logistical operations. Liu et al., [24] discussed AI integration into smart logistics CPS is set to transform how items are moved, stored, and delivered, resulting in a more efficient, secure, and sustainable logistics landscape.

The proliferation of Internet of Things (IoT) devices in smart logistics has created a slew of security issues, demanding the creation of strong cyber-secured frameworks to protect the integrity and confidentiality of logistical operations. Device authentication, secure communication routes, data encryption, access control methods, intrusion detection systems, and vulnerability management tactics should all be included in a comprehensive cyber security framework. This multimodal strategy safeguards sensitive data, inhibits unauthorized access, and mitigates cyberattacks, establishing a secure and resilient smart logistics ecosystem [25].

The seamless integration of Internet of Things (IoT) devices into intelligent logistics-based cyber-physical systems (CPS) has transformed supply chain management, increasing efficiency and visibility. However, the growing density of IoT devices offers substantial issues in assuring bonded coverage and connectivity, both of which are critical for reliable data transmission and efficient logistics operations. The seamless integration of Internet of Things (IoT) devices into intelligent logistics-based cyber-physical systems (CPS) has transformed supply chain management, increasing efficiency and visibility. However, the growing density of IoT devices offers substantial issues in assuring bonded coverage and connectivity, both of which are critical for reliable data transmission and efficient logistics operations. To solve these issues, Abbas, and MArwat [26] proposed a scalable simulated frameworks are a great tool for analysing and optimising IoT device location and communication tactics.

2.3. Machine learning approaches

The researchers proposed expanding the study to include all five-day traffic log files and constructing a consensus-based ML model. On the KDD Cup 1999 dataset, [27] used feature reduction approaches combining Information Gain (IG) and Chi-Square (CR) with a J48 classifier. With only 16 features, the reduced feature set surpassed the original, earning a 99.84 % detection rate. To increase performance in the future, the authors recommend investigating ensemble feature selection or other filter-based strategies. [28] suggested a supply chain logistics cyber security framework. The framework is intended to safeguard the supply chain against a wide range of cyber threats, such as malware attacks, data breaches, and supply chain interruptions The Cyber Vulnerabilities Detection System (CVDS) analyses network traffic and device behaviour for anomalies that may signal a cyber intrusion or assault, using advanced machine learning and artificial intelligence algorithms. ML approaches were utilised by Saghezchi et al., [29] to detect DDoS attacks in Industry 4.0 CPPSs. They tested 11 ML approaches on network data from a semiconductor manufacturing industry and discovered that supervised algorithms including DT, RF, and KNN performed better at detecting network traffic patterns.

2.4. Deep learning (DL) approaches

To enhance IoT security, [8] Proposes a deep learning system for detecting DoS attacks in IoT networks. A deep-learning algorithm developed by the researchers can recognize dangers posed by denial-of-service (DoS) attacks. This research made use of the Python programming language as well as the scikit-learn, Tensorflow, and Seaborn packages. The outcomes demonstrated that the deep learning model significantly increased accuracy, making IoT network threat prevention more effective. [11] proposes a unique approach for rapid and accurate attack detection in IoT networks that combines deep learning and three-level algorithms. The methodology showed noticeably better detection performance than previous methods when tested on the BoT-IoT dataset. This method is also versatile, which makes it a promising addition to IoT security because it may be used to improve of other IoT applications security. A hybrid deep learning (DL) CNN-BiLSTM model was developed to detect DDoS attacks using benchmark data [12]. The model obtained an accuracy of up to 94.52 percent during training, testing, and validation using the CIC-DDoS2019 dataset by picking the most relevant features based on their scores in the provided dataset. CIC-DDos2019 dataset was applied by [30] on a deep learning model to detect DDoS attacks on network traffic. On the CIC-DDos2019 dataset, the DNN model obtained an amazing 95 % accuracy. By collecting network traffic from virtual machines and Internet of Things (IoT) devices, the researchers want to produce a new dataset that is equivalent to CIC-DDos2019 in the future.On the CICIDS 2017 dataset. Using Deep Learning, Aldhyani et al., [31] developed a model with CNN and LSTM using the CIC-DDOS2019 dataset, and then improved it by including the CICFlowMeter-V3 network. Their proposed approach was completely accurate in all evaluation metrics. In the future, the model can be used with custom data. Mohammadian et al., [32] proposed adversarial attacks for deep learning intrusion detection, assessing their model on the CIC-IDS2017, CIC-IDS2018, and CIC-DDoS2019 datasets. Their findings revealed that by utilising smaller features in adversarial sample creation, the suggested approach outperformed competing attacks. They investigated the model utilizing success rates of the best feature sets, adversarial class average confidence, and adversarial sample transferability.

2.5. Ensemble learning approaches

Sambangi et al., [33] investigated an ensemble ML model employing feature selection with information gain and regression analysis. They concentrated on the log files from Friday morning and afternoon with Benign, Bot, and DDoS classes. The ensemble model had a 97.86 % accuracy for the Friday morning dataset and a 73.79 % accuracy for the Friday afternoon dataset. Using DT and RF classifiers, Li et al., [27] suggested an ensemble tree-based ML IDS approach. It improves attack detection and predictions by providing explanations. The datasets namely, NF-BoT-IoT-v2, IoTDS20 as well as NF-ToN-IoT-v2, were utilised in the evaluation.

2.6. Research gap

Despite the success of machine learning and deep learning in a variety of applications, the problem of picking appropriate features still needs to be solved. [11] presented ensemble-based method for forecasting logistics-based IoT-based vulnerabilities using benchmark data. Traditional deep learning classifiers, on the other hand, may need to be better in predicting logistics-based IoT-based risks with adequate feature selection. We provide a workable hybrid (LSTM + CNN) DL model with selected feature to enhance logistics-based IoT-based vulnerability detection in order to overcome these constraints.

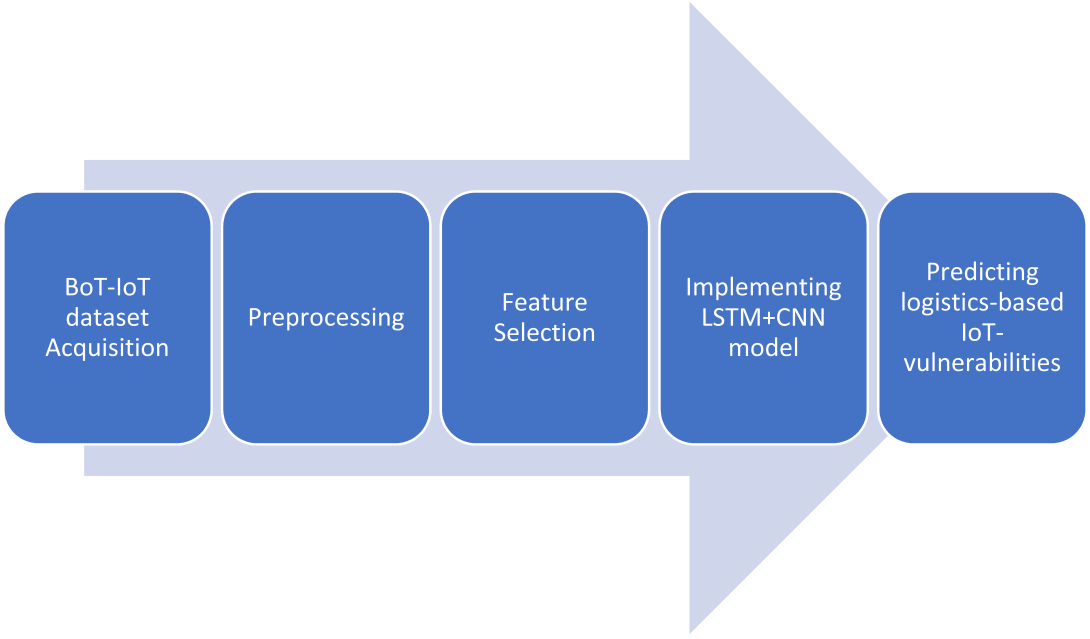

Fig. 2. Summary of the suggested system.

Table 2
Detail about the dataset.

Sno.	Vulnerabilitymain category	Vulnerability sub category	# of records
1	Information Gathering	Service scanning	1,463,364
		OS Finger Printing	358,275
		Total	1,821,639
2	Denial of Service	DDoS	38,532,480
		DoS	33,005,194
		Total	71,537,674
3	Information Theft	Keylogging	1469
		Data theft	118
		Total	1,587
Grand Total73,360,900			

3. Methodology

The suggested method consists of four major modules (Fig. 2): (i) acquisition of BoT-IoT datasets, (ii) preprocessing, (iii) feature selection, and (iv) implementation of the proposed (LSTM + CNN) model. Each module is explained in depth below.

3.1. BoT-IoT dataset acquisition

Researchers [8,11] have employed a variety of datasets to evaluate their systems, and the BoT-IoT dataset generated by UNSW Canberra was used in this study.

What is the BoT-IoT dataset? The BoT-IoT dataset contains network traffic captures from both normal and botnet traffic. It was developed by the UNSW Canberra Cyber Range Lab to offer researchers with a realistic dataset for developing and evaluating intrusion detection systems (IDS) and intrusion prevention systems (IPS) for IoT networks. The primary objective of this dataset is to aid in the research and development of effective intrusion detection and cybersecurity systems for IoT environments [18].

The dataset covers botnet assaults such as denial-of-service (DoS), distributed denial-of-service (DDoS), command-and-control (C&C) communication, and data exfiltration. Normal traffic, such as online browsing, email, and file sharing, is also included in the dataset [10].

The BoT-IoT dataset is a great resource for researchers working on innovative intrusion detection and prevention systems (IDS and IPS) for IoT networks. It can be used to assess the efficacy of various detection methods as well as to find potential vulnerabilities in IoT devices and networks. The BoT-IoT dataset can be downloaded for free from the UNSW Canberra Cyber Research Group website. The files in the dataset are labelled and categorised based on attack categories and subcategories [18]. The dataset shows a realistic network environment with both botnet traffic and typical traffic. It includes DDoS (Distributed Denial of Service) attacks based on TCP, UDP, and HTTP protocols, as well as DoS (Denial of Service) attacks based on the same protocols. Furthermore, the dataset covers attacks related to information collecting, such as Service Scanning and OS Fingerprinting, as well as attacks connected to information theft, such as Keylogging and Data theft. The dataset, which contains 73,360,900 records in ".csv" format, is used to train and test machine learning and deep learning models. Table 2 contains further information about dataset.

3.2. Preprocessing

Before training deep learning models, the dataset is preprocessed to improve training appropriateness and reduce overfitting. The following techniques are used for preprocessing:

The dataset must be preprocessed before training deep learning models to improve training appropriateness and decrease overfitting. Preprocessing consists of the following steps:To extract the dataset from its MySQL tabular form, which includes concatenated records from successive tables with the labelling process, we used an auto-incrementing function called 'pkSeqID.' The query'select * from IoT Dataset UNSW 2018 into outfile '/path/to/file.csv' was then used. Fields were terminated by ',' and lines were terminated by 'n';' to convert the dataset into CSV format. Because CSV format is widely supported for such reasons, this conversion allows us to analyze the data more easily using multiple Python modules and facilitates easier sharing of the dataset. Furthermore, due to the large size of the created dataset, which contained over 72,000,000 records and took up around 16.7 GB in CSV format, with a corresponding pcap size of 69.3 GB, data management

proved rather difficult. To address this issue, we chose to take a 5 % representative sample from the original dataset using the same select MySQL queries discussed before. This subset, referred to throughout this paper as the training and testing sets, consists of four files totaling around 0.78 GB in size and contains approximately 3 million data.

3.2.1. Handling missing value

To resolve the dataset's missing values, we used a data modelling approach to infer these values. To fill in the missing numbers, we generated simple column statistics such as the arithmetic mean. Furthermore, in circumstances where some protocols (such as ARP) resulted in the absence of source and destination port numbers (thus rendering them inapplicable), we assigned these values as −1, indicating an invalid port number. This step was critical for the dataset's appropriate evaluation. To make statistical methods more easily used, we turned categorical feature values in the dataset into consecutive numeric values. For example, categorical values in the 'state' property such as 'RST', 'CON', and 'REQ' were translated to '1′, '2′, and '3′, respectively. This change made subsequent steps of our investigation easier to analyze and manipulate data [31].

3.2.2. Normalization

Normalization is the process of uniformly scaling all columns in a dataset. When the property ranges across columns differ greatly, this is required. There are several approaches for achieving normalization in Machine Learning, and we used the minimum-to-maximum scale method in our work. In this procedure, from the highest value the lowest value is subtracted in each column, as well as the result is split by the range of the column. As a result, the modified columns' minimum and maximum values will be 0 and 1. This normalization step is critical for statistical models and deep learning approaches to effectively converge and overcome challenges with local optima. This method of normalization is known as Min-Max Normalization (see Equation (1)):

$$x\_norm = (x - min) / (max - min) \quad (1)$$

where:

x is the initial value, the normalized value is x_norm, min is the dataset's lowest value, and max is the dataset's maximum value.

Furthermore, certain protocols (e.g., ARP) resulted in the absence of source and destination port numbers in the dataset, rendering them inapplicable. We assigned these missing data the value −1, which represents an invalid port number, to ease accurate assessment. In addition, we converted the dataset's categorical feature values into consecutive numeric values to facilitate the application of statistical methods. Categorical values in the 'state' property, for example, 'RST', 'CON', and 'REQ' were mapped to '1′, '2′, and '3′, respectively. This conversion facilitated data analysis and processing in following stages of our research.Furthermore, when importing the dataset, we used a shrinking technique to assure a randomized sample by randomly removing records. During the cleaning process, we replaced occurrences of 'infinity' with '-1,' and any rows containing 'NaN' entries were eliminated. In addition, nine attributes having a constant value of '0′ were removed, resulting in the model being trained with 69 attributes.

3.2.3. Labelling

Table 2 describes how the class tags were sorted into categories. The label 'BENIGN' was assigned a value of '0′ to detect network activity attacks, while other attack types (information gathering, denial of service, and information theft) were labelled with values '1′, '2′, and '3′, respectively.

3.2.4. Encoding

Encoding is a technique used in an ML model to convert categorical variables into numerical values, allowing them to be used in the model. The following steps are involved in this process: Tokenization of each element in the document, remove all punctuation, all category variables encoding, and transform the document in to a sequence.

Table 3
A partial list of optimal features.

F.No	Optimal Features	Optimality Score
Of1	AR_P_Proto_Dport	0.72
Of2	AR_P_Proto_P_DstIP	0.68
Of3	AR_P_Proto_P_Sport	0.64
Of4	AR_P_Proto_P_SrcIP	0.55
Of5	drate	0.53
Of6	flgs number	0.48
Of7	N_IN_Conn_P_DstIP	0.39
Of8	N_IN_Conn_P_SrcIP	0.31
Of9	Pkts_P_State_P_Protocol_P_SrcIP	0.26
Of10	Pkts_P_State_P_Protocol_P_DestI	0.19

3.2.5. Splitting module

The dataset was gathered and prepared in three distinct sets: (i) Training Set, (ii) Validation Set, and (iii) Testing Set. This section aided in model training, hyper parameter adjustment, and unbiased evaluation of the model's performance on real-world data.

3.2.5.1. Training set. The training set serves as the foundation for training a model. It is essential in learning model parameters like weights and biases, which are required for the model's predictive capabilities. We set aside 90 % of the dataset for training in our technique. This training set contains both input and output data, which allows the model to understand the correlations and patterns in the data and hence enhance its forecasting abilities.

3.2.5.2. Validation set. The validation set contains data that is distinct from the training set and is used to assess and validate our model's performance during training. This validation approach gives us valuable information for fine-tuning our hyperparameters, such as training or learning relevant parameters, model type, and architecture. It is used to address the issues of under fitting and overfitting in our model. We set aside 10 % of the data for validation, allowing us to optimize the model's performance and confirm its generalizability to previously unseen data.

3.2.5.3. Testing set. Distinct collection of data is the testing set that utilized to estimate the model of training performance on real-world data. It provides an objective evaluation of how well the final model operates in practice. The model is then used to predict the output for the data in the testing set once it has been trained and validated using the training and validation sets. This ensures that the model's capabilities are properly tested on previously unknown data. In our technique, we set aside 10 % of the data in the dataset for testing purposes.

3.3. Feature selection

Rather than selecting all of the variables from the source data, we focus on identifying the most critical features for detecting logistics-based IoT risks. To do this, different approaches, including as principal component analysis (PCA) [21], decision trees [32]., and the chi-squared (x2) test [33], can be used to extract the most optimal features from raw data.

The study [12] used an x2 test to rank and choose traits, showing good results. The x2 test evaluates class and feature dependency using correlations between predictor and target variables. The computation procedure is as follows.

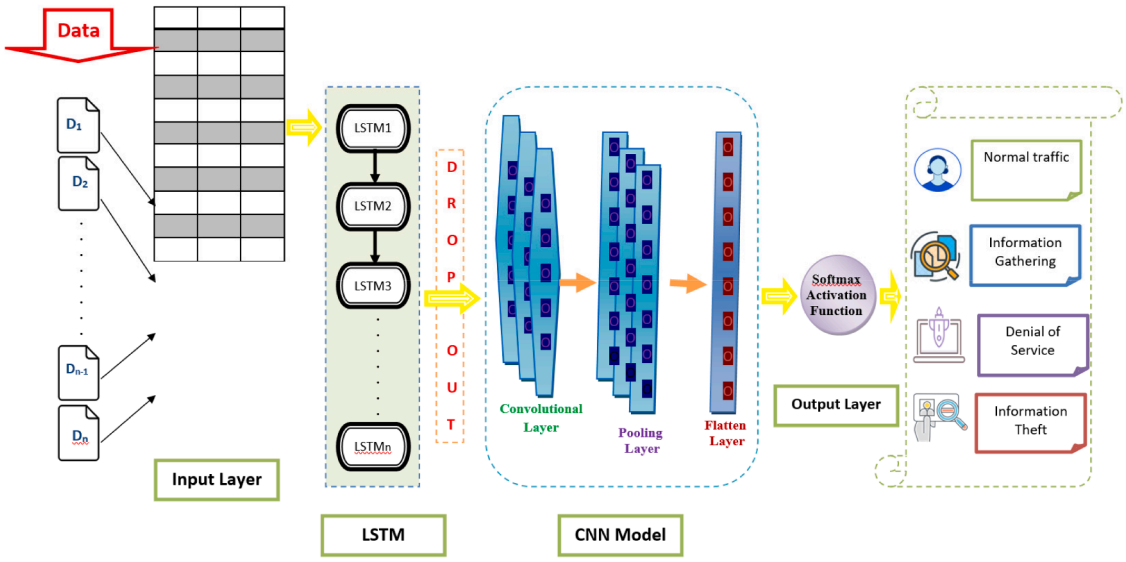

Fig. 3. Deep learning model based on LSTM + CNN for detecting vulnerabilities in logistics-based IoT data exchange.

$$Y_c^2 = \sum \frac{(Ai - Bi)^2}{Bi} \tag{2}$$

The x2 test was used to the original dataset to pick significant characteristics with strong links to the target variables, based on degree of freedom (c), observed value (A), and predicted observation (B). The Python-based Sklearn package, which incorporates the Select KBest score and the Chi2 function, was used to generate more optimal features with higher association to the target characteristic. A portion from 24 attributes of the BoT IoT dataset were used to build the learning model. Table 3 lists the top ten most essential features based on their relationship to attribute values, arranged by importance and reliance on the class that targeted.

3.4. Implementing composite deep learning model for vulnerabilities detection in IoT data exchange

A Summary of the Proposed Model: The suggested LSTM + CNN model for detecting and classifying IoT data exchange vulnerabilities works as follows: Input data embedding converts input into numeric data, LSTM layer retains long-term dependencies, Dropout layer prevents overfitting, CNN performs feature extraction, Pooling layer reduces feature map dimension, Flatten layer unrolls the map of pooled feature in to the feature vector, and Softmax function classifies input text into emotion categories. Fig. 3 depicts the model workflow. Each layer's detailed information is provided below.

3.5. Implementing proposed model

Following preprocessing, the LSTM + CNN deep learning model is used to classify input data into multiple IoT-based vulnerability categories. Layers in the model include Embedding, LSTM, Dropout, Convolutional, Maxpooling, Flattening, and Output. Fig. 4 depicts the workflow of the LSTM + CNN model.

3.5.1. Embedding layer

In this work, by the use of Keras embedding layer the vector of embedding data has created. The embedding layer created a feature matrix i.e embedding matrix with two-dimensional using the formula $D \in \mathscr{R}^{t \times n}$, where'$D$' shows data input,'$R$' depicts real no.,'$t$' represents the length of the input data and 'n' denotes the dimension of a input embedding. After constructing the embedding matrix, it was passed on to the next layer for additional processing.

3.5.1.1. LSTM Layer. LSTM layer is the next component of our proposed strategy. It takes the matrix of embedding as input and uses four LSTM gates to execute various computations: the forget gate, input gate, candidate gate, as well as an output gate. These gates work together to allow the LSTM layer to properly learn long-term dependencies. However, in practise, larger LSTM models are frequently restricted to operate within a time step range of 250 to 500. Equations 1 through 6 below constitute the foundation for the LSTM model.

$$F_t = \sigma_g\left(W_f \times x_t + U_f \times h_{t-1} + b_f\right) \tag{2}$$

$$i_t = \sigma_g(W_i \times x_t + U_i \times h_{t-1} + b_i) \tag{3}$$

$$o_t = \sigma_g(W_o \times x_t + U_o \times h_{t-1} + b_o) \tag{4}$$

$$c'_t = \sigma_c(W_c \times x_t + U_c \times h_{t-1} + b_c) \tag{5}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t \tag{6}$$

$$h_t = o_t \cdot \sigma_c(c_t) \tag{7}$$

Table 4 shows list of abbreviations used in LSTM model.

3.5.2. Dropout layer

The main reason for including this layer is to solve the issue of overfitting. The "rate" parameter, represented by the value 0.5, within the range of 0 and 1. The Dropout layer selectively deactivates either at random"drops out" neuron activations within the LSTM layer by applying dropout to it. With the dropout rate set to 0.5, approximately 50 % of the neurons will be deactivated during training. The dropout layer successfully prevents neuron co-adaptation by making neurons more independent of one another. This promotes the model to learn numerous useful representations for the same input, which reduces the possibility of overfitting. The dropout layer is normally turned off during the testing phase, and the whole model with all neurons activated is used to make predictions [12].

The dropout modeling for a particular neuron is depicted in Equation# 7.

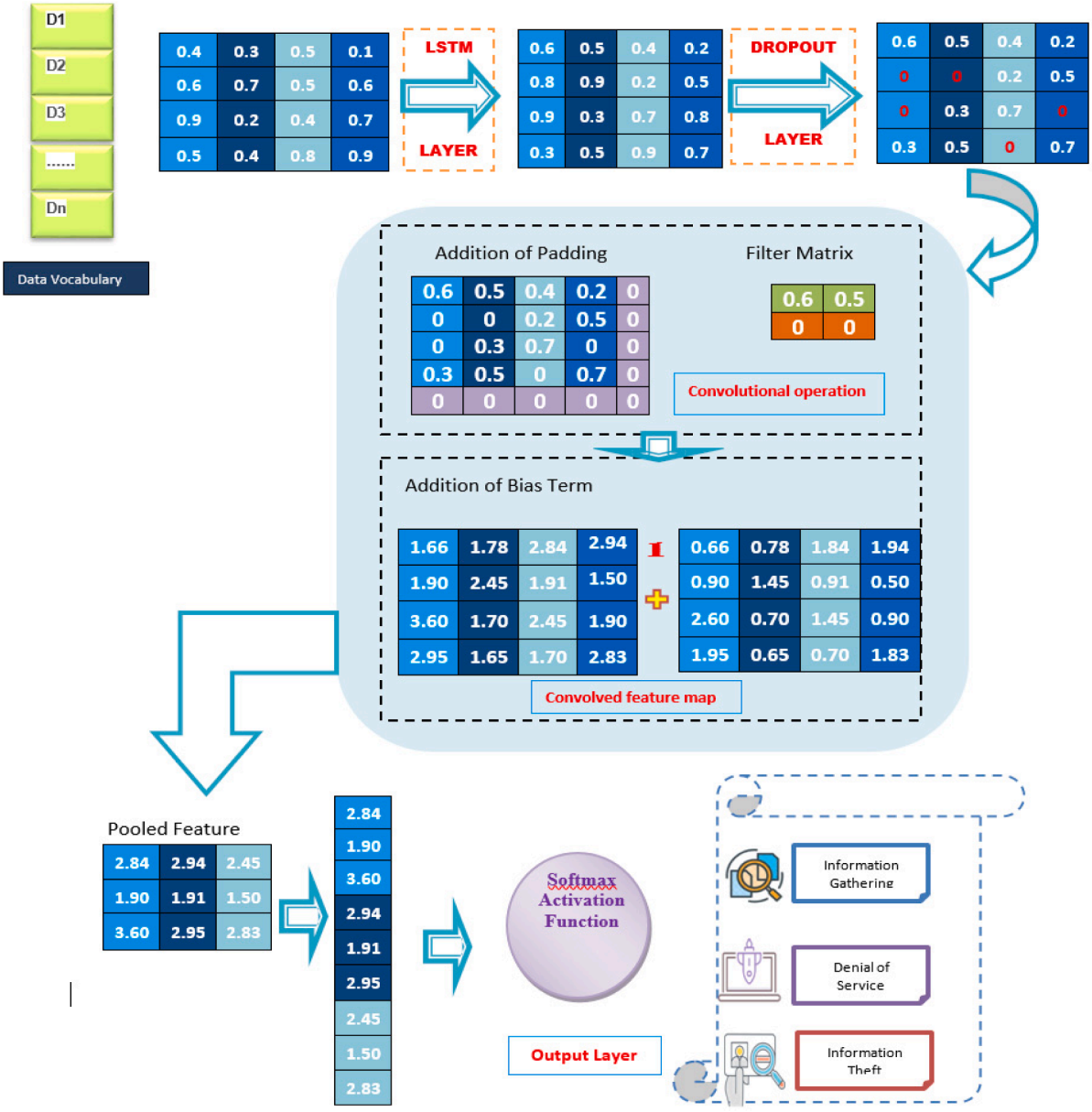

Fig. 4. Detailed working of the proposed model.

Table 4
List of abbreviations of LSTM Model.

List of abbreviations	Description
F_t	Forget gate
i_t	Input gate
o_t	Output gate
c'_t	Candidate gate
c_t	Cell state
h_t	Hidden state
σ_g	Sigmoid function
σ_c	hyperbolic tangent function
$W_f W_i$, W_o, $W_c U_f$, U_i, U_o, U_c	Weights for the respective gates
x_t	Vector size of input gate
h_{t-1}	preceding output
b_f, $b_{i,}$, b_o, b_c	Biases

$$F(r, s) = \begin{cases} r & if\ s = 0 \\ r - 1 & if\ s = 1 \end{cases} \quad (8)$$

The variable "r" reflects the desired results in the context stated, as "s" indicates the probability aspect of the representation of real number input. When "s" equals 1, the neuron containing the true value is dropped or deactivated, whereas any other value of "s" (not equal to 1) indicates that the neuron stays engaged.

3.5.3. Convolutional layer

A convolutional operation is performed in this layer, which involves a mathematical operation applied to two functions to produce a third function. The dimensions about input matrix (N), filter matrix (T), as well as output matrix (O) are represented like following to conduct this operation:

$$I = \mathbb{R}^{MxN} \tag{9}$$

The LSTM layer generates the input matrix I, as defined by Equation (9). $\mathbb{R}$ denotes the collection of all real numbers,*M* the length, and *N* the width of the input matrix.

$$F = \mathbb{R}^{mxn} \tag{10}$$

The filter matrix is represented as F in Equation (10), where $\mathbb{R}$ indicates the collection of all real numbers. The variables *m* as well as *n* indicate the length and width of the filter matrix.

$$O = \mathbb{R}^{w \times x} \tag{11}$$

The output matrix is denoted as O in Equation (11), where $\mathbb{R}$ denotes the set of all real numbers. The variable w represents the length, while the variable x represents the width of the output matrix.

When given an input data or feature map (matrix) I of size M N and a filter (also referred to as a kernel or weight matrix) F of size m n, the convolution operation is carried out by sliding the filter over the input matrix and computing element-wise multiplication followed by a summation of the interconnecting elements.

The convolutional operation is denoted by Equation (12):

$$Output\,(i,j) \;=\; Æ©_{p=1}^{m} Æ©_{q=1}^{n} I(i+p-1, j, q-1) \bullet F(p,q) \tag{12}$$

Where:

$Output(i,j)$ is the element of the output feature map located at position (i,j).

$I(i+p-1,j,q-1)$ is the input feature map element at point.$I(i+p-1,j,q-1)$.

$F(p,q)$ is the filter element at position (p,q).

The filter's dimensions are *m and.n*

The dimensions of the input feature map are denoted by *M and.N*

Each element of the output matrix is merged with the relevant bias term after the convolutional process. For example, adding the bias value to the first element of the output matrix yields $0.64 + 1 = 1.64$, which represents the value of the first element in the feature map for the given input data.This layer, as shown in Algorithm No. 1, employs the following parameters: (i) "filters": The amount of filters used in the convolutional layer is represented by this parameter; (ii) "kernel_size": The dimensionality of the convolutional window is indicated by this parameter; (iii) "padding": This parameter has three options: "valid," "same," or "casual"; When the "padding" property is set to "valid," no padding is applied; When "padding" is set to "same," the output length is guaranteed to be the same as the original input length. When "padding" is set to "casual," dilated convolution is facilitated; and (iv) "activation = relu" is a parameter: This indicates that the ReLU activation function was used to introduce nonlinearity.Consider 1.64 is an initial element for specific data for input in map feature. After applying activation function relu, $Output = max(0, 1.64)$, that converts $Output = 1.64$ since 1.64 is bigger than 0. This method is repeated for each feature map element, resulting in additional rectified feature map elements for the provided data.

3.5.4. Pooling layer

. This layer's main objective is to decrease the dimensionality of each feature map while keeping essential information intact. The pooling layer is used to lessen the computational load. By aggregating information, the pooling layer reduces the feature map dimensionality. In turn, every data stream is dominated to max pooling in on dataset that selected the highest value for representing the main features about the data input. Equation (13) provides a mathematical definition of the max pooling process.

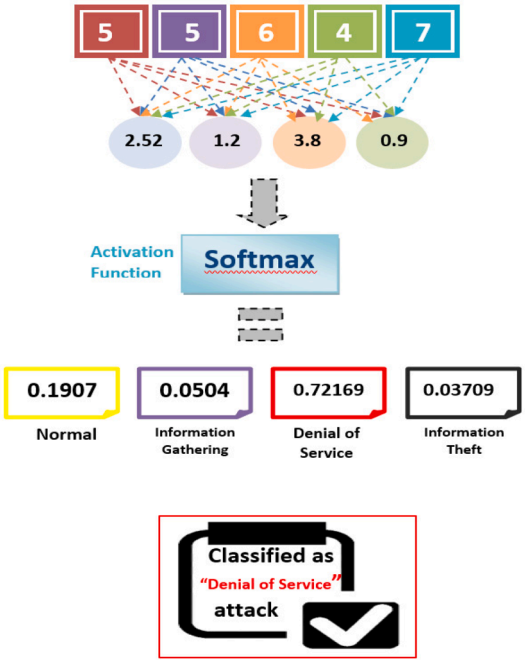

Fig. 5. Classification of logistics-based IoT vulnerabilities using the softmax.

$$Oij = \max(I(2i-1)(2j-1), I(2i-1)(2j), I(2i)(2j-1), I(2i)(2j)) \tag{13}$$

Where:

The output value at point (i,j) in the pooled feature map is represented by the symbol.*Oij*

$I(2i-1)(2j-1), I(2i-1)(2j), I(2i)(2j-1),$ and$I(2i)(2j))$ are the original feature map input values at the corresponding places in a 2*x*2 window.

The max pooling operation chooses the highest value within each window of size 2x2 and stores it as the output at that place. This procedure aids in shrinking the spatial dimensions of the feature map while maintaining the most important elements.Eq. (13) showsthata window of size (22) is placed over the feature map to construct the pooled feature map, and the maximum element within the window is extracted. In this situation, for example, the greatest element, 2.23, corresponds the initial element within the pooled feature map generated from the provided data input using the selected window scale (maximum of 1.64, $1.76, 1.80,$ *aswellas*2.23). A corresponding process is used to process the remaining data in the pooled feature map.

3.5.5. Flatten layer

This layer in the CNN converts the pooled feature map in a column vector that serves as the input for the neural network's classification task. A column vector representation modified to the target text is obtained from feature map. The flattened presentation of the pooled feature map is accomplished using reshaping function of numpy, as depicted in the equation#14. This reshaping function concatenates the feature vectors to generate the column vector representation.

$$\text{Flattening} = pooled.reshape(f-w+1)\,x\,(v-h+1) \tag{14}$$

The given equation concatenates all the rows of the feature map, including row 1, row 2, row 3, and so on, into a single-column vector.

3.5.5.1. Output Layer. The CNN output layer is responsible for making

the final predictions based on the learned features from the previous layers. In the case IoT-based vulnerabilities classification, the output layer typically consists of neurons corresponding to each class label in the dataset. The output of each neuron represents the probability of the input belonging to its respective class.

In our IoT-based vulnerability categorization problem, we have C = 4 categories. C = 4 neurons will be used in the output layer, designated as O1,O2,…,OC. Each neuron's output is calculated with a softmax function to ensure that the values are between 0 and 1 and that the sum of all output values is 1.

To use the softmax function, we must first assess the cumulative input, as given in Equation (15).

$$t_i = \sum w_i l_i + b \tag{15}$$

Whereas '*w*' be regarded as the "weight vector", '*l*' mean the "input vector", as well as '*b*' speaks for "bias factor". The function softmax is used to determine the probability of the various classes or labels for the supplied input vector 't'.

$$softmax\,(t_i) = \frac{e^{t_i}}{\sum_{n=1}^{m} e^{t_n}} \tag{16}$$

where "e" is the base of the natural logarithm (about equivalent to 2.71828) and ti is the raw output of the ith neuron prior to applying the softmax.

The softmax function has the following mathematical definition:

The output of each neuron after applying the softmax function represents the probability of the input belonging to its associated class. The class with the highest likelihood is used to make the final forecast for the input.

3.6. Test example

This section delves into the intricate arithmetic used to forecast IoT-related vulnerabilities based on the available historical data. The detailed explanation of the composite model's functions provides insight into its operations.

To determine the probability associated with each tag, the SoftMax function is employed, such as *"t1", "t2", "t3", "t4"*, and so on, using the final output of the model as input. The net input was calculated using Equation (16), which improved the model's ability to predict.

For example, if we have four classes (C = 4) and the raw outputs of the neurons are t1 = 2.5, t2 = 1.2, t3 = 3.8, and t4 = 0.9, we get the following results after applying the softmax function (Eq. 16):

$$softmax(t_1) = \frac{e^{t1}}{e^{t1}+e^{t2}+e^{t3}+e^{t4}} = \frac{e^{2.5}}{e^{2.5}+e^{1.2}+e^{3.8}+e^{0.9}} = \frac{12.91829}{67.714263} = 0.1907.$$

$$softmax(t_2) = \frac{e^{t2}}{e^{t1}+e^{t2}+e^{t3}+e^{t4}} = \frac{e^{1.2}}{e^{2.5}+e^{1.2}+e^{3.8}+e^{0.9}} = \frac{3.41490}{67.714263} = 0.0504.$$

$$softmax(t_3) = \frac{e^{t3}}{e^{t1}+e^{t2}+e^{t3}+e^{t4}} = \frac{e^{3.8}}{e^{2.5}+e^{1.2}+e^{3.8}+e^{0.9}} = \frac{48.86879}{67.714263} = 0.72169.$$

$$softmax(t_4) = \frac{e^{t4}}{e^{t1}+e^{t2}+e^{t3}+e^{t4}} = \frac{e^{0.9}}{e^{2.5}+e^{1.2}+e^{3.8}+e^{0.9}} = \frac{2.51209}{67.714263} = 0.03709.$$

So, in this example, the final prediction is the IoT-based vulnerability class with the highest probability, which is class t3 = 0.72169.

As a result of this investigation of historical traffic data, the anticipated IoT-based vulnerability result has been classified as "denial of service" (Fig. 5).

To summarize, the CNN output layer uses the softmax function to the raw outputs of its neurons to generate a probability distribution over the classes, allowing the model to make correct predictions.

Algorithm 1 displays the suggested model's pseudocode methods for forecasting IoT-based Vulnerabilities.

Algorithm 1: IoT-based Vulnerabilities Detection

1. INPUT: BoT-IoT(Processed data) tagged dataset **D** as csv file
2. Divide utilizing Scikit learn in train (S_{train}, NR_{train}) −test (S_{test}, NS_{test})
3. Create vocabulary for integer mapping to BoT-IoT
4. Convert every data stream for BoT-IoT into integer sequences
5. **Process** model LSTM-CNN X-train, Y-train
PARAMETER SETUP:
i. max_features = 80000, **input_length** = 53, **embed_dim** = 128,
ii. Classes = 5, **Epoch** = 7, **batch_size** = 8
iii. LSTM units equal to 100
iv. CNN: - filters equal to 16; pool size equal to 2, Kernel size equal to 8
#Create Deep learning model with different layers
model←Sequential ()
#Embedding Layer
Add (embedding(max-features, embed-dim, input-length equal to max-len))to the model
#Convolutional Layer
model.add (Conv1D (filters, kernel_size, padding = 'same', activation = 'relu')
model.add (Conv1D (filters, kernel_size, padding = 'same', activation = 'relu')
#Dropout Layer
model.add (Dropout (0.5))
#Maxpooling Layer
model.add (MaxPooling1D (pool size))
#Flatten Layer
model.add (Flatten ())
#Dense/Activation Layer
model.add (Dense (classes, activation = 'softmax'))
#Compile Function
model. compile (loss = categorical_crossentropy, optimizer = adamax, metrics = [accuracy])
#Model Summary
print (model. summary ())
for all epochs in (1: Epoch) do
#Fit model on train data
model.fit (X_train, y_train, epochs, batch_size = batch_size, validation_data = (X_test, y_test))
End for
Accuracy = model. evaluate (X_test, y_test, verbose = 2, batch_size = batch_size)
Output: return Accuracy
End Procedure

```
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models

import Sequential
from tensorflow.keras.layers import LSTM, Dense, Conv1D, MaxPooling1D, Dropout, Flatten

# Load the BoT-IoT dataset
data = pd.read_csv('botnet_iot_traffic.csv')
```

Fig. 6. Loading BoT-IoT dataset.

Table 5
Model parameterization for LSTM-CNN.

Model/Layer	Parameter and Values
LSTM + Other(Pooling etc.)	Embedding dimension = 300 Batch size = 8,16,32 Number of Epochs = 10 Dropout = 0.5 Pooling = max pooling Filter Size = 8,10 Unit size = 10,15,20,30,50,60,80,100,130,150 Activation function = softmax
CNN	No. of convolutional layers = 1 Filer sizes = 2,4,6,8,11 Padding='same' Activation function=='Relu' Kernal Sizes = 2,3,4

4. Experiments and their outcomes

We did Python experiments towards 2.0 TensorFlow as well as Keras platforms to evaluate the proposed framework's efficiency and efficacy. A machine with an Intel Core i7-7700 CPU and 32 GB of RAM was used to conduct the experiments.

How was the BOT-IoT Dataset used in this research? The dataset, which contains 73,360,900 entries in ".csv" format, is used to train and test the proposed deep learning model.Using the BoT-IoT dataset with an LSTM + CNN model to categorise network types entails multiple phases, beginning with data loading. This procedure is outlined in Python-based steps below: (See Fig. 6).

4.1. Answer to research question no. 1

We used multiple LSTM-CNN models to classify logistics-based IoT vulnerabilities into distinct classes by adjusting various parameters in each layer to answer the first research question, "*how accurately does the LSTM + CNN Composite DL model forecast* logistics-based *IoT vulnerabilities when given a benchmark dataset*?" We attempted to construct the best effective LSTM-CNN model on the benchmark dataset by using a parameter tweaking technique inspired by [24], which focused on IoT-based vulnerability detection.

Table 5 shows the model trainable parameters that were refined during this approach. Using multiple LSTM-CNN models, we classified logistics-based IoT vulnerabilities into three classes using the recommended deep model with parameter changes. Several experiments were run using various parameter configurations. Based on the results of multiple trials with various settings, the suggested LSTM-CNN model combines the most optimal combination of parameters (refer to Table 5).

Table 6 shows the construction of 10 LSTM-CNN models with varied parameter combinations, such as filter numeral, filter scale, pool capacity, as well as LSTM layer units. The filter number settings are kept constant between 3 and 12, as their values have no major impact on the model's efficiency.

We examined accuracy(test), loss(test), and training time after running numerous tests on several LSTM-CNN models with varying parameters (see Table 7).

LSTM-CNN (10) model with 16 filters and a filter size i.e 8 has exceeded than other models through accueacy of 78 percent. Established that in the time of testing increasing the size of the filter expanded model's training time.

4.1.1. Complexity of the proposed algorithm

The purpose is to compute temporal complexity of the algorithm (LSTM + CNN). For achieving, we concentrate on assessing the time complexity of the LSTM and convolutional layers. Since LSTM's spatial as well as temporal local, storage requirements of network doesn't effected by the length of the input, resulting in a time complexity of O(1) per weight at every time unit. In essence, the overall time required to create an LSTM $O(w)$, where w represents the "number of weights". The complexity of CNN can be estimated using the equation O(kj = 1 xj − 1. p2j. xj. y2j) [34]. The constants k represents the quantity of convolutional layers, xj is the number of filters that in the j^{th} layer, x input

Table 6
Combinations of parameters for various LSTM + CNN models.

Model	Parameter setup
LSTM-CNN (1)	Filters = 16 LSTM unit size = 80 Filter size = 8
LSTM-CNN (2)	Filters = 10 LSTM unit size = 10 Filter size = 11
LSTM-CNN (3)	Filters = 16 LSTM unit size = 60 Filter size = 8
LSTM-CNN (4)	Filters = 16 LSTM unit size = 150 Filter size = 8
LSTM-CNN (5)	Filters = 16 LSTM unit size = 15 Filter size = 11
LSTM-CNN (6)	Filters = 16 LSTM unit size = 130 Filter size = 8
LSTM-CNN (7)	Filters = 16 LSTM unit size = 50 Filter size = 8
LSTM-CNN (8)	Filters = 16 LSTM unit size==150 Filter size = 8
LSTM-CNN (9)	Filters = 16 LSTM unit size = 15 Filter size = 11
LSTM-CNN (10)	Filters = 16 LSTM unit size = 20 Filter size = 11

Table 7
Test Accuracy(TA), Test Loss(TL), and Training Time(TT) for different LSTM + CNN models.

Model	TA (%) and TL (%)	TT(s)
LSTM-CNN (1)	TA = 63 TL = 1.09	15 s
LSTM-CNN (2)	TA = 65 TL = 1.35	2 s
LSTM-CNN (3)	TA = 68 TL = 1.02	16 s
LSTM-CNN (4)	TA = 70 TL = 1.07	14 s
LSTM-CNN (5)	TA = 73 TL = 1.04	3 s
LSTM-CNN (6)	TA = 76 % TL = 1.07	10 s
LSTM-CNN (7)	TA = 78 % TL = 0.93	8 s
LSTM-CNN (8)	TA = 80 TL = 0.92	19 s
LSTM-CNN (9)	TA = 81 TL = 0.79	6 s
LSTM-CNN (10)	TA = 83 % TL = 0.90	7 s

Table 8
Comparison of efficiency (A = Accuracy, P = Precision, R = Recall, as well as F = F-score) of proposed model against ML models.

Methods/classifier	A(%)	P(%)	R(%)	F(%)
SVM	71	73	72	72
KNN	67	66	67	67
DT	65	63	62	64
RF	77	75	74	75
XGB	58	59	58	58
LR	72	76	72	71
Our Approach (LSTM + CNN)	95.73	96.64	94.15	93

channel numbers in the j^{th} layer, *Pj* the spatial size of the filter, and *Yj* the spatial scale of the output feature map.The computational cost of (LSTM plus CNN) per clock cycle can be approximated as the sum of the LSTM layer and the convolutional layer complexity: O(kj = 1 xj − 1. p2j. xj. y2j + w). It denotes that the overall complexity of the model may be written as O(w) when using normal asymptotic notation.

4.2. Answer to research question no. 2

To answer the second research question, "How does the proposed LSTM + CNN model compare with other ML and DL methods?" we evaluated the LSTM + CNN model done for detection of IoT-basedvulnerabilities as well as compared it to the performance of classical machine learning techniques and other deep neural networks. Table 8 displays the results of performance evaluations for machine learning methods, deep neural networks, and the proposed LSTM-CNN model.

- **SVM and LSTM + CNN (Proposed)**

In comparison to the recommended model, SVM (Support Vector Machine) performed worse in detecting IoT-based vulnerabilities (accuracy = 71 %, precision = 73, recall = 72, f-score = 72). The need for feature scaling hampered SVM performance, resulting in lower accuracy [20]. Furthermore, SVM is not well-suited for huge datasets and may struggle when coping with increased noise and overlapping target classes. SVM's performance declines when the number of features for each data set exceeds the number of training data samples. Furthermore, determining the suitable kernel function for SVM can be difficult.

- **KNN and LSTM + CNN(Proposed)**

The proposed model i.e LSTM + CNN is measures in an experiment with machine learning model KNN (K-nearest neighbours). Table8 depicts evaluation of machine learning model results, with accuracy = 67percent, precision = 66, recall = 66, and f-score = 67. KNN's inferior accuracy can be related to its constraints in dealing with huge, imballanced datasets through high-dimensions. KNN at risk for noisy data, information that is null, as well as outliers, all of which degrade its performance. To perform successfully, feature scaling and homogenised elements are required [12].

- **DT and LSTM + CNN (Proposed)**

The Decision tree efficiency is evaluated with the recommended model i.e LSTM + CNN. The experimental outputs reveals that the Decision tree model has not performed well, with an accuracy of 65percent, precision of 63, recall of 62, and f-score of 64.The DT model's poor performance can be due to its extended training time, which reduces its effectiveness. Furthermore, the DT model's capacity to estimate future values is limited. It can only produce a single attribute, and overfitting can occur when subjected to noise in the dataset [11].

- **RF and LSTM + CNN(Proposed)**

in order to compare the results of RF i.e random forest model with our proposed system we get the performance shown in table 8 that accuracy equal to 77 %, precision = 75, recall = 74, and f-score = 75). The RF model's poor performance could be attributed to its slower prediction and longer learning time, resulting in worse overall efficiency.

- **XGB and LSTM + CNN(Proposed)**

The performance of XGB (Extreme Gradient Boosting) is compared with proposed system in this section.

for detecting IoT-based vulnerabilities. The XGB classification model yielded the following results: accuracy = 58 %, precision = 59, recall = 58, and f-score = 58. The long training and scoring processes of the XGB classifier contribute to its low accuracy, which reduces its overall efficiency. Furthermore, as XGB only worked on numerical featues due to this XGB is hard to change as well as the results may be overfitted if its weights not correctly set.

- **LR and LSTM + CNN(Proposed)**

The less accuracy has generated by LR when it is compared with proposed model. (accuracy = 72 %, precision = 76 %, recall = 72 %, and f-score = 71 %).Because of its proclivity to generate overfitting, LR yielded unsatisfactory outcomes. Before using LR, it is critical to remove outliers [35]. Furthermore, LR performs poorly with non-linear input data and may oversimplify real-world situations by assuming a linear relationship between variables.

- **RNN and LSTM-CNN (Proposed)**

In the experiment, RNN (Recurrent Neural Network) was compared against our proposed model. RNN outperforms the recommended model with 57 % accuracy, 64 % precision, 62 % recall, and 64 % F-score. RNN's lesser accuracy can be linked to its limits in capturing long-term dependencies, which has an impact on its performance. RNNs frequently suffer fading and bursting gradients, which reduces their accuracy [12].

- **CNN and LSTM-CNN (Proposed)**

In this experiment, we compared the performance of CNN to our proposed model. Our proposed approach outperformed the CNN model in this comparison. With 69 % accuracy, 76 % precision, 74 % recall, and 74 % F-score, Table 9 demonstrates the CNN model's subpar

Table 9
Comparison of efficiency (A = Accuracy, P = Precision, R = Recall as well as F = F-score) of proposed model against DL models.

Model	A(%)	P(%)	R(%)	F(%)
RNN	57	64	62	64
CNN	69	76	74	74
LSTM	67	75	71	73
BiLSTM	70	76	63	64
CNN + RNN	66	69	60	58
Proposed (LSTM-CNN)	95.73	96.64	94.15	93

performance.The CNN model's poor performance can be due to its proclivity for overfitting, which reduces accuracy. Furthermore, CNN requires a large dataset to work successfully and struggles when given a smaller dataset [21].

- **LSTM and LSTM-CNN(proposed)**

An experiment was carried out to compare the performance of the proposed LSTM-CNN model to that of the LSTM model, the LSTM model's efficiency was low, with an accuracy of 67 %, precision of 75 %, recall of 71 %, and f-score of 73 %.The LSTM model's low efficiency can be due to its unidirectional nature, which only stores information from the past. As a result, the LSTM cannot capture and store future information, resulting in decreased efficiency.

- **BiLSTM and LSTM-CNN (Proposed)**

The preliminary analysis compared the BiLSTM deep learning approach to the proposed LSTM-CNN model for detecting IoT vulnerabilities. According to the testing results, the effectiveness of the BiLSTM model was found to be low, with an accuracy of 70 %, precision of 76 %, recall of 63 %, and f-score of 64 %.

- **CNN-RNN and LSTM-CNN (Proposed)**

Finally, the proposed model for IoT vulnerabilities detection is compared to a CNN-RNN model combination. The CNN-RNN technique produced unsatisfactory results, as indicated in Table 9. When tanh is used as the activation function, RNNs have difficulty interpreting extremely long sequences. As a result, RNNs are unsuitable for building highly deep learning models since they struggle to handle extensive dependencies [12].

The Evaluation Metrics i.e Accuracy, Precision, Recall as well as F-score for various Deep Learning Models are depicted in Table 9.

4.3. Performance measures

In this section, we use the confusion matrix to evaluate our model's learning performance. The confusion matrix is divided into four sections: false positive (FP), false negative (FN), true positive (TP), and true negative (TN). By displaying the confusion matrix, we can demonstrate our model's classification performance by identifying correct and incorrect predictions. Fig. 7 shows that values in each cell in this figure represent the actual count for each class in the sample confusion matrix. For example, there were 120 "Normal" class instances that were correctly classified as "Normal" (True Positives), 50 "Information Gathering" class instances that were incorrectly classified as "Normal" (False Positives), 30 "Denial of Service" class instances that were incorrectly classified as "Normal" (False Positives), and 40 "Information Theft" class instances that were incorrectly classified as "Normal" (False Positives). For each class and prediction combination, the other cells have similar actual counts.

Furthermore, our proposed model is evaluated using standard intrusion detection system metrics. Equations (8)-(10) offer mathematical equations for precision, recall as well as F-score.

Accuracy: Equation (17) computes accuracy, providing information about the model's prediction ability. It measures the ratio of correctly identified alerts to both correctly detected as well as misclassified alarms created by the IDS model to determine IDS's success. The following equation determines it:

$$Accurcay = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

$TP = truepositive$, $TN = truenegative$, $FP = falsepositive$, $andFN = falsenegative$

Precision, commonly known as the false negative rate (FNR), is the proportion of misclassified attacks to total attack cases. It determines how many positive forecasts are correct. Equation (18) is used to calculate precision:

$$Precision(p) = \frac{TP}{FP + TP} \tag{18}$$

$p = precision$, $TP = truepositive$, $FP = falsepostive$, $andFN = falsenegative$

Recall:The true positive rate (TPR) or detection rate (DR) defines the proportion of correctly identified malicious events among all malicious occurrences.The recall is calculated using Equation (19) and indicates how many true positives are effectively predicted:

$$Recall(r) = \frac{TP}{FN + TP} \tag{19}$$

$r = recall, TP = truepositive, andFN = falsenegative$

F-score:Since it accounts for both false positives and false negatives,

	Actual: Normal	Actual: Information Gathering	Actual: Denial of Service	Actual: Information Theft
Predicted: Normal	120	50	30	40
Predicted: Information Gathering	20	100	10	30
Predicted: Denial of Service	10	30	90	20
Predicted: Information Theft	15	40	25	80

Fig. 7. A snapshot of sample confusion matrix.

Table 10
Performance comparison of LSTM + CNN models both with and without choosing features[FS(0) = beyond selection on features, FS(1) shows along with selection on features, A = Accurcay, P = Precision, R = Recall, F = F1-score]

Model Name	Accuracy (%)		Precision (%)		Recall (%)		F1-Score (%)	
	FS (0)	FS(1)	FS (0)	FS(1)	FS (0)	FS(1)	FS (0)	FS(1)
LSTM + CNN–1	83	89.13	79	88	75	86	76	87
LSTM + CNN −2	86	90.01	81	89	77	87	76	88
LSTM + CNN −3	89	90.62	74	89	72	88	69	88
LSTM + CNN −4	84	91.66	72	90	66	89	65	89
LSTM + CNN −5	83	93.57	79	90	74	89	75	89
LSTM + CNN −6	82	92.48	80	93	75	90	76	92
LSTM + CNN −7	83	93.31	72	93.61	65	93	68	92.71
LSTM + CNN −8	87	93.86	84	95.37	78	93	79	93.16
LSTM + CNN −9	81	94.21	77	96.31	73	93.83	75	94.46
LSTM + CNN −10	89	95.73	83	96.64	80	94.15	81	95.52

Table 11
Proposed Model's A: Accuracy (%), P:Precision (%), and R:Recall (%)in ablation Study.

Model ID	Model	A(%)	P(%)	R(%)
M1	LSTM + CNN (without considering FS)	89	83	80
M2	LSTM + CNN (with FS)	95.73	96.64	94.15
M3	CNN Only (without LSTM)	82	81	80
M4	LSTMOnly (without CNN)	78	80	79

the F1 score is essential in determining how well the proposed systems work. With regards to coping with unbalanced class label distributions, this statistic is particularly helpful. Equation (20), which calculates the F1 score, shows how recall and sensitivity are balanced, highlighting the general consistency of the model's predictions.

$$F_{score} = 2x\frac{P \text{ x } R}{P + R} \tag{20}$$

R = Recall, P = Precision

Table 10 shows the accuracy, recall, as well as F-score of various LSTM + CNN models along with and in the absence of feature selection. Our suggested LSTM + CNN (10) model achieved the greatest accuracy of 95.73 % when the parameters filter number = 16, filter size = 8, and unit size = 100 were used. Furthermore, the proposed model achieved promising results in terms of precision (96.64 %), recall (94.15)%, and F1-score (93 %).

Rationale for improved results: To attain better results, we recommend combining LSTM with CNN. The efficient forward and backward storage of context data by LSTM adds to enhanced performance. CNN improves data representation by preventing information degradation and allowing more accurate court decision prediction. The capability of anomaly detection approaches is demonstrated by the hybrid DL with feature selection. In terms of accuracy, recall, F1-measure, and precision, our technique surpasses shallow learners. DL algorithms excel in detecting intrusions accurately, overcoming the limits of traditional classification that rely on standard feature encoding for anomaly detection.

4.3.1. Ablation study

To evaluate the efficiency of each module, an ablation research on LSTM + CNN was performed. Four scenarios were investigated: one without feature selection (model 1), one without LSTM (model 3) and one without CNN (model 4). Table 11summarizes the outcomes as well as extra performance analysis.

When LSTM + CNN was compared to model 1, the removal of the FS module resulted in lower accuracy because the approach couldn't use the best features for classification. The usefulness of LSTM + CNN was clear when compared to model 3, as its performance worsened when the LSTM subsystem was removed, resulting in the loss of contextual data. Furthermore, removing the CNN component caused the model to lose its capacity to capture local characteristics in the data stream, demonstrating the importance of CNN in overall model performance.

4.4. Cross validation

To evaluate the efficiency of the suggested method, the following experiment comprises randomizing the dataset that utilizing validation strategy i.e k-fold. In this work, dataset is separated into k equal-sized subgroups for analysis, specifically using a set of 10 folds as mentioned in reference [12].

The data is divided into ten equal folds with N = 10, resulting in different subsets. As shown in Table 12, each subset is subjected to cross-validation testing, utilizing for training9 folds and for testing 1 fold. This methodical methodology evaluates the suggested method's robustness and accuracy.

Several classifiers were evaluated using a 10-fold cross-validation technique. Table 13 displays information such as mean accuracy, standard deviation of the mean, mean precision macro, standard deviation of precision macro, mean recall macro, standard recall macro, and mean F1

Table 12
A list of randomized 10-fold cross validation samples.

N-Fold(N = 10)	D = Dataset (1–1000)									
	1	2	3	4	5	6	7	8	9	10
N1	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N2	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N3	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N4	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N5	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N6	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N7	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N8	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N9	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000
N10	1–100	101–200	201–300	301–400	401–500	501–600	601–700	701–800	801–900	901–1000

Table 13
Comparing cross-validation of the suggested approach to other classifiers.

Model	Accuracy (mean)	Standard Deviation	Macro Precision (mean)	Standard Deviation	Macro Recall (mean)	Standard Deviation	MacroFI (mean)	Standard Deviation
SVM	78	0.05	81	0.06	82	0.07	82	0.05
KNN	76	0.05	75	0.07	74	0.07	75	0.07
RF	86	0.06	85	0.05	84	0.06	85	0.06
CNN	88	0.06	88	0.06	87	0.06	88	0.07
LSTM	89	0.07	89	0.06	88	0.05	89	0.06
Proposed(LSTM + CNN)	95	0.04	96	0.05	94	0.04	95	0.05

Table 14
Substantial distinctions between the RF (ML) and LSTM + CNN models.

	Accurate Classification with RF	Mis-classification with RF	Total:
Accurate classification LSTM + CNN	62	4	66
Misclassification with LSTM + CNN	14	20	34
Total	76	24	100

We reject the null hypothesis and accept the alternative hypothesis based on the chi-squared value of 2.1, p-value of 0.133, and 1 degree of freedom, which shows statistical significance for both models.

macro scores. Notably, the suggested LSTM + CNN model outperformed all other tested classifiers, yielding the best results.

4.5. Significance testing

Two experiments were carried out to determine the importance of the study (see Tables 14 and 15). It was not by chance that the LSTM + CNN (DL) model outperformed the RF (ML) model statistically. The tests involved choosing 100 records at random from the dataset and categorizing them with LSTM + CNN (DL) and RF (ML) classifiers. In this experiment, two hypotheses were investigated.

Hnull: The error rates in both models are the same.

Halternate: Both models' error rates are significantly different.

McNemar's chi-squared statistic test is represented by Equation (21).

$$\chi^2 = (|x - y| - 1)^2/(x + y) \quad (21)$$

Discordant test statistics were computed using cells x and y, with 1 degree of freedom and χ^2 signifying chi-squared.

4.5.1. Analysis

Table 10 shows the astounding 95.73 % accuracy of the LSTM + CNN model in predicting IoT-related vulnerabilities from historical traffic data. Table 8 demonstrates, however, that the RF model performs poorly across all evaluation metrics. Through statistical analyses, a substantial difference between the DL (LSTM + CNN) and ML (RF) models was found, demonstrating LSTM + CNN's superiority. The LSTM + CNN model's resilience in addressing IoT vulnerabilities was strengthened by the use of word embeddings, underscoring its advantages in the study of historical traffic data.

4.6. How does the LSTM + CNN model achieve such high accuracy in identifying IoT vulnerabilities?

LSTM-Based Sequential Learning: The model's LSTM component is critical in capturing sequential dependencies in the data. Given the dynamic nature of IoT environments, where data streams are frequently time-dependent, LSTM excels at recognizing and retaining long-term patterns. This allows the model to detect tiny variations in the behaviour of IoT devices that may signal possible vulnerabilities over time.

Using CNN for Spatial Feature Extraction: CNN, renowned for its spatial feature extraction capabilities, complements LSTM by capturing local patterns and spatial dependencies in the IoT data. The hierarchical feature extraction in CNN allows the model to discern intricate details in the input data, enhancing its ability to recognize specific characteristics associated with vulnerabilities

Efficient Blending of Temporal and Spatial Information: The integrated combination of temporal and spatial characteristics made possible by the integration of CNN and LSTM enables a comprehensive analysis of IoT data. Due to this synergy, the model can extract pertinent information from the spatial configurations and temporal sequences of IoT devices, which facilitates a thorough knowledge of potential vulnerabilities.

Generalization and Adaptive Learning: The LSTM + CNN model has adaptive learning capabilities, allowing it to alter its parameters in response to the changing nature of IoT data. This adaptability improves the model's generalisation performance, allowing it to detect vulnerabilities across a wide range of IoT scenarios and settings.

Noise and Variability Resistance: The model is robust to noise and variability that are frequent in IoT data. The model's robustness in the face of real-world IoT data difficulties is enhanced by the combination of LSTM's capacity to filter out irrelevant information and CNN's robust

Table 15
Comparison of the proposed model with the benchmark findings.

Work and Model	Efficiency in the controlled experiments with different datasets (A:Accuracy, P:Precision, R:Recall)			Reported Efficiency
	BoT-IoT	CICDDoS(2019)	CIC-IDS(2017)	
Alghazzawiet al. [12]Deep Learning model	A(85),P(86), R(85), F(86)	A(87),P(78), R(87), F(87)	A(81),P(82), R(83), F (82)	A(92),P(91), R(90), F (91)
Alosaimi, S., & Almutairi, S. M[11]ML Classifier (Ensemble)	A(73),P(73), R(74), F(73)	A(75),P(76), R(75), F(76)	A(78),P(79), R(78), F (79)	A(85),P(86), R(87), F (86)
Alzahrani & Asghar [16]	A(82),P(80), R(81), F(80)	A(80),P(79), R(78), F(80)	A(82),P(82), R(81), F (82)	A(89),P(87), R(88), F (88)
Our Approach (LSTM + CNN + with enhanced FS)	A = 95.73P = 96.64 R = 94.15F = 95.52	A = 94.23P = 95.04 R = 93.12F = 94.51	A = 93.02 P = 91.54 R = 92.04 F = 92.18	N/A

feature extraction procedures.

4.7. Answer to research question no. 3

This section addresses RQ3, which compares benchmarks to the suggested logistics-based IoT vulnerability prediction approach. Direct comparisons are difficult and have limited future application due to issues like dataset variations and imprecise methodology in other studies.

The specified difficulties that suggested before, we set out to execute the approaches presented in the linked research papers utilizing two datasets. Our goal was to accurately replicate the original studies and methodology described in the papers. However, due to a lack of details and incomplete conversations in some circumstances, we were forced to make assumptions or overlook certain components of the technique. For example, in one of the articles [11] a supervised ML model for predicting IoT-based vulnerabilities using historical traffic was proposed. On the historical traffic data, they used a ensemble learning technique. Unfortunately, the experimental results obtained on the given benchmark dataset showed poor model performance, with accuracy, precision, recall, and F1-score all reporting 75 %. However, no precise data or specifications were supplied in the publication, making it difficult to replicate their exact approach. Despite these challenges, we made every effort to follow the available information and execute the strategy to the best of our abilities.

In our study, we used three cutting-edge datasets from [11,12] to conduct a comprehensive quantitative evaluation of various IoT-based vulnerability detection systems. The use of an Anaconda-based Jupyter notebook was used to implement recognized approaches [12]. We discovered some inconsistencies between our results and the provided results during the review process. Varying results are due to the utilization of different datasets, parameter combinations, as well as software tweaks. To illustrate, Alghazzawi et al., [12] claimed an accuracy of 92 percent in their study, whereas our studies showed a 85 percent accuracy. Similarly, [16,12] reported accuracy scores of 85 and 89 percent in their investigations, whereas our testing on the BoT-IoT dataset produced 73 and 82 percent accuracy. The authors' use of various datasets with varying properties can explain the variations in stated and observed accuracies. Furthermore, [16] used a DL method in the study focusing on supply chain traffic data, which may have contributed to the varied conclusions. Our thorough review method took into account all elements to assure the dependability and correctness of our findings. When enhanced feature selection is combined with a DL model, its effectiveness is increased. [11] used an ensemble ML approach to develop an ML model that predicted BoT-IoT attacks, however it underperformed due to insufficient features. In contrast, our LSTM + CNN hybrid DL model outperforms prior techniques, thanks to enhanced feature selection, implying that different DL model combinations should be explored for future BoT-IoT attack prediction, since it has the potential to produce even more promising results. Our DL-based solution excels in forecasting BoT-IoT attacks, with significant influence from well-selected predictor parameters, demonstrating its efficacy in predicting IoT vulnerabilities. It is built on a hybrid deep neural network and better feature selection.

4.7.1. Results discussion

The suggested Cyber Vulnerabilities Detection System with LSTM + CNN model in Logistics-based IoT Data Exchange showed numerous critical insights into the system's behavior and robustness.

The Influence of Hyperparameters:

- The learning rate has a substantial impact on the model's convergence speed and accuracy. A higher learning rate resulted in faster convergence but overfitting, whereas a lower learning rate resulted in slower convergence but better generalization performance.
- The batch size affects the model's training efficiency by determining the amount of samples handled in a single iteration. A bigger batch size increased training speed but may result in poor performance, whereas a smaller batch size decreased training speed but increased generalization.
- The number of epochs, which represents the total number of iterations over the training data, has an effect on the model's capacity to learn patterns from the data. Increasing the number of epochs improves accuracy in general, but could lead to overfitting.
- The number of LSTM units in the LSTM layer affects the model's capacity to capture long-term dependencies in IoT data. Increased layer size may improve the model's capacity to learn temporal patterns.
- The granularity of spatial feature extraction was dictated by the filter size in the CNN layers. Changing the size of the filter could improve the model's capacity to extract relevant spatial data.
- Dropout, a regularization strategy, avoided overfitting by removing neurons at random during training. The dropout rate was adjusted to assist in limiting the model's complexity and increase generalization.

The Implications of Data Quality

- The presence of missing values in IoT data may affect the model's performance. Maintaining accuracy required accurate input of missing values.
- Data outliers can have a negative impact on the model's performance. Identifying and dealing with outliers aided in improving the model's robustness.
- The raw IoT data is transformed into a range between 0 and 1, making it easier for the LSTM + CNN model to understand significant patterns and correlations within the data. This may improve the accuracy with which cyber vulnerabilities are detected.
- Normalized data can facilitate a faster convergence of the LSTM + CNN model during training by reducing the need for the model to cope with significant fluctuations in the data scale. This can' considerably cut training time while also improving overall efficiency.
- Normalization can prevent overfitting, which occurs when a deep learning model memorizes the training data too well and fails to generalize to new data. Normalized data makes it more difficult for the model to overfit, resulting in improved generalization performance.

Accuracy: The proposed system identified cyber risks in logistics-based IoT data with an overall accuracy of 95.73 %. The system's ability to properly extract relevant characteristics from the data and use these features to train the LSTM + CNN model accounts for the system's high accuracy. The LSTM + CNN model can recognize both temporal and spatial relationships in IoT data, which is critical for detecting cyber vulnerabilities accurately.

Scalability: To meet the growing number of IoT data in logistics operations, the proposed system is scalable. The modular nature of the system enables for simple integration with additional data sources and processing units. The LSTM + CNN model can potentially be trained on larger datasets to increase its accuracy.

Commentary on comparative results: In detecting cyber vulnerabilities in IoT data, the LSTM + CNN model outperformed classic machine learning models such as support vector machines (SVMs) and decision trees. This is due to the LSTM + CNN model's capacity to capture the data's complex temporal and spatial correlations, which is required for accurate vulnerability detection.

4.8. Generalizability of experimental findings

To assess the generalizability of the suggested logistics-based IoT-based vulnerability prediction system, we carried out a thorough study. The following elements were carefully looked at:

Collecting data: The quality and the diversity of the training and

test data determine how generalizable the system is. It captures different risk patterns and contextual components by utilizing a variety of benchmark data, improving generalizability across logistics-based IoT architectures, and time periods.

Model creation and algorithm choosing: Selecting appropriate modelling strategies and algorithms is necessary to guarantee generalizability. The proposed model architecture and algorithm were determined to be suitable for reflecting the complexity and dynamic of logistics-based IoT-based vulnerability identification based on the literature research and our testing.

Performance evaluation: Selecting appropriate modeling strategies and algorithms is necessary to guarantee generalizability. The proposed model architecture and algorithm were determined to be suitable for reflecting the complexity and dynamic of logistics-based IoT vulnerability identification based on the literature research and our testing.

4.9. Potential applications of this approach in the logistics industry, and how could it improve overall system performance?

Detecting Anomalies in Supply Chain Operations: The model's LSTM component, which may capture temporal dependencies, can be used to analyze historical data relevant to supply chain processes. The model can successfully identify anomalies or deviations by learning the normal patterns of activities and transactions, signalling potential risks such as unauthorized access, tampering, or discrepancies in shipment routes.

Fleet Management Using Predictive Maintenance: The model can analyse time-series data from vehicle sensors, maintenance logs, and previous performance measures by leveraging LSTM's sequential learning capabilities. This allows for the anticipation of probable equipment breakdowns or maintenance requirements. Logistics firms may reduce downtime, enhance fleet efficiency, and optimise maintenance schedules by proactively addressing concerns.

Warehouse Security: A Spatial Analysis: The CNN component of the model excels at extracting spatial features. When applied to warehouse surveillance video feeds, it can detect unusual patterns or actions that may reveal security risks, such as unauthorised worker access or strange movements. This improves overall warehouse security and lowers the likelihood of theft or tampering.

Detecting Cybersecurity Threats in Logistics Networks: The model can be used to analyse network log data and communication patterns inside logistics networks. The capacity of LSTM to capture temporal dependencies aids in the detection of odd network behaviours, whereas CNN can detect spatial patterns suggestive of potential cyber threats. This improves the logistics network's cybersecurity defences and aids in the prevention of unauthorized access or data breaches.

Improved Security Personnel Decision Support: Because of the model's accurate vulnerability detection capabilities, security personnel can receive real-time insights and alerts. This enables them to respond quickly to possible threats, immediately investigate occurrences, and adopt preventive measures, boosting the overall responsiveness and effectiveness of security operations.

In conclusion, the proposed LSTM + CNN model has a wide range of applications in the logistics business, from anomaly detection to predictive maintenance and cybersecurity. Logistics firms may improve overall system performance, decrease risks, and ensure the safe and efficient transportation of goods across the supply chain by leveraging their superior analytical capabilities

4.10. Justifying proposed approach for the technology research area

Meeting an Important Need: To ensure the seamless flow of goods, the logistics industry primarily relies on IoT data sharing. The increased integration of IoT devices in logistics operations increases the exposure to cyber threats. The proposed architecture directly addresses the crucial requirement for a strong cyber vulnerability detection system customised to the logistics sector's particular issues.

Higher Level Analytical Communication abilities: LSTM + CNN integrates the advantages of Convolutional Neural Network (CNN) for spatial analysis and Long Short-Term Memory (LSTM) for sequential learning. This combination of sophisticated analytical tools enables the model to thoroughly examine time-series data from Internet of Things (IoT) devices, detecting both spatial abnormalities and temporal trends linked to possible cyber threats.

Customised for Logistics-Specific Challenges: Logistics operations provide distinct problems, such as varying environmental conditions, dynamic routes, and a diversified spectrum of IoT devices. By merging sequential learning to capture temporal dependencies and spatial analysis to extract features important to the logistics context, the proposed model is specifically tailored to address these issues. Because of this, it is a focused solution for the logistics area.

Preventative Threat Mitigation: The LSTM + CNN model enables logistics organizations to take proactive efforts to prevent potential dangers by recognizing cyber vulnerabilities in real time. This proactive strategy is critical in an era where cyber threats are continually developing, and prompt response is required to avert supply chain disruptions.

Making a Difference in the Cybersecurity Landscape: By providing a specialized solution for the logistics sector, the proposed approach contributes to the larger cybersecurity landscape. As logistics operations become more digitized and interconnected, the demand for domain-specific cybersecurity solutions grows. By tackling the unique issues provided by logistics-based IoT data transmission, the LSTM + CNN model adds to the developing field of cybersecurity research.

Generalization Potential: The LSTM + CNN model, while created for the logistics industry, may have broader ramifications and the possibility for generalisation to other IoT-intensive areas. The research on this model can provide insights into the development of successful cybersecurity solutions for a variety of industries, enhancing the general understanding of IoT security.

In conclusion, the proposed LSTM + CNN model for Cyber Vulnerabilities Detection System in Logistics-based IoT Data Exchange justifies its place in technology research by directly addressing industry-specific challenges, providing advanced analytical capabilities, and contributing to the evolving field of cybersecurity with practical applications and generalizability. This study is in line with the increasing demand for safe and resilient IoT ecosystems in logistics and related industries.

5. Conclusions and future work

While digital technology has rapidly altered many industries, the logistics industry has taken a more cautious approach to innovation. As a result, rapid vulnerability assessment and vigilant system monitoring are critical for efficient breach response. Cybersecurity must become a strategic objective for maintaining transportation and logistics (T&L) safety standards. In this work, we developed a LSTM + CNN Composite DL model for cyber vulnerabilities detection in logistics-based IoT data exchange. The LSTM + CNN hybrid deep neural network model was employed in the research to identify and categorize vulnerabilities. Dataset acquisition, preprocessing, feature selection, and classification were all parts of the proposed system. Using a variety of datasets, including pertinent BoT-IoT attack categories, we trained and assessed the model. When compared to conventional ML approaches, feature selection improved accuracy (95.73 %), precision (96.64 %), recall (94.15 %), and F-score (95.52 %) by prioritising the most pertinent features.

5.1. Internet of things data exchange technology

IoT Data exchange technology enables data interchange between IoT devices and various applications and platforms.

This technology is vital for real-time data-driven decision-making,

optimizing supply chain operations, and improving overall logistics efficiency. Following are the key benefits IoT data exchange: (i) Increased visibility and tracking of goods and assets across the supply chain, (ii) Allows for real-time monitoring of inventory levels, transportation status, and potential disruptions, enabling for proactive measures to rectify concerns and optimize operations, (iii) Reduces delivery times and costs by facilitating optimised route planning and delivery schedules based on real-time data, and (iv) Improves overall supply chain visibility and efficiency by enabling secure data exchange and collaboration among stakeholders in the supply chain.

As IoT usage of devices grows, IoT Data Exchange technology will become increasingly important for managing complicated supply chains and guaranteeing efficient and safe logistics operations.

Limitations: However, the suggested system has several shortcomings, including: (i) The identification of significant features (predictors) using only one statistical method, the chi-squared measure; Not utilizing DL models that have already been trained (pertained).

Future Work:(i) We intend to assess its performance in future using more dataset, (ii) In addition to the chi-squared analysis, we want to investigate additional feature selection technique; (iii) Currently, each attack class is categorized separately. Our research will develop to incorporate a multi-class classification system at a more granular level in the future, and (iv) Furthermore, researchers looking into logistics-based IoT vulnerability screening may find it helpful to use the BoT-IoT dataset in combination with hybrid DL. Due to its increased accuracy, the hybrid DL model with enhanced feature selection is a great choice for intrusion detection and safeguarding software-based networks.We intend to increase the security of logistics-based-based IoT devices by enlarging the spectrum of network attacks to include a wider range. The conversion of an CVDS into an cyber veulnerability prevention system (CVPS) is our goal.

Funding

This research work was funded by Institutional Fund Projects under grant no. (IFPIP: 325–611-1443). The authors gratefully acknowledge technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

CRediT authorship contribution statement

Ahmed Alzahrani: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Supervision, Project administration, Funding acquisition. **Muhammad Zubair:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability statement

The data underlying this work can be requested from the corresponding author.

Acknowledgments

The authors gratefully acknowledge technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

References

[1] Latif MNA, Aziz NAA, Hussin NSN, Aziz ZA. Cyber security in supply chain management: a systematic review. LogForum 2021;17(1):49–57.
[2] Prabhughate A. Cybersecurity for Transport and Logistics Industry. Infosys: Bengaluru, India; 2020.
[3] Cybersecurity in the logistics industry. (n.d.). Krontech.com. Retrieved August 14, 2023, from https://krontech.com/cybersecurity-in-the-logistics-industry.
[4] Boyson S, Corsi TM, Paraskevas JP. Defending digital supply chains: Evidence from a decade-long research program. Technovation 2022;118:102380.
[5] Nasir MA, Sultan S, Nefti-Meziani S, Manzoor U. (2015, June). Potential cyber-attacks against global oil supply chain. In 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA) (pp. 1-7). IEEE.
[6] Sarker IH, Khan AI, Abushark YB, Alsolami F. Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. Mobile Networks and Applications 2022:1–17.
[7] Bhardwaj A, Kaushik K, Bharany S, Rehman AU, Hu YC, Eldin ET, et al. IIoT: traffic data flow analysis and modeling experiment for smart IoT devices. Sustainability 2022;14(21):14645.
[8] Susilo B, Sari RF. Intrusion detection in IoT networks using deep learning algorithm. Information 2020;11(5):279.
[9] Vishwakarma R, Jain AK. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. Telecommun Syst 2020;73(1):3–25.
[10] Azath H, David DB, Blessie EC, Jayapradha A, Rani SS. (2021, November). BoT-IoT based Denial of Service Detection with Deep Learning. In 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 221-225). IEEE.
[11] Alosaimi S, Almutairi SM. An intrusion detection system using BoT-IoT. Appl Sci 2023;13(9):5427.
[12] Alghazzawi D, Bamasag O, Ullah H, Asghar MZ. Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. Appl Sci 2021;11(24):11634.
[13] Ferrag MA, Shu L, Djallel H, Choo KKR. Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0. Electronics 2021;10(11):1257.
[14] Asghar MZ, Rahman F, Kundi FM, Ahmad S. Development of stock market trend prediction system using multiple regression. Comput Math Organ Theory 2019;25:271–301.
[15] Skrodelis HK, Romanovs A. In: October). Cyber-Physical Risk Security Framework Development in Digital Supply Chains. IEEE; 2021. p. 1–5.
[16] Alzahrani A, Asghar MZ. Intelligent risk prediction system in IoT-Based supply chain management in logistics sector. Electronics 2023;12(13):2760.
[17] Ahmad S, Asghar MZ, Alotaibi FM, Awan I. Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. HCIS 2019;9:1–23.
[18] Sobb T, Turnbull B, Moustafa N. Supply chain 4.0: a survey of cyber security challenges, solutions and future directions. Electronics 2020;9(11):1864.
[19] Parker S, Wu Z, Christofides PD. Cybersecurity in process control, operations, and supply chain. Comput Chem Eng 2023;108169.
[20] Saba T, Rehman A, Sadad T, Kolivand H, Bahaj SA. Anomaly-based intrusion detection system for IoT networks through deep learning model. Comput. Electr. Eng. 2022, 99, 107810. [CrossRef].
[21] Chagas DA, Rocha Filho GP, Meneguette RI, Bonacin R, Gonçalves VP. In: May). Machine Learning for Detection of Distributed Denial-of-Service Attacks from Queries Executed in DBMS. SBC; 2023. p. 57–70.
[22] Sarder MD, Haschak M. Cyber security and its implication on material handling and logistics. College-Industry Council on Material Handling Education 2019;1(1):1–18.
[23] Zhan J, Dong S, Hu W. Ioe-supported smart logistics network communication with optimization and security. Sustainable Energy Technol Assess 2022;52:102052.
[24] Liu Y, Tao X, Li X, Colombo A, Hu S. Artificial intelligence in smart logistics cyber-physical systems: state-of-the-arts and potential applications. IEEE Transactions on Industrial Cyber-Physical Systems 2023.
[25] Abbas AW. Cyber Secured Framework for Control and Monitoring of IoT Devices in Smart Logistics. University of Engineering & Technology Peshawar (Pakistan)); 2021. Doctoral dissertation.
[26] Abbas AW, Marwat SNK. Scalable emulated framework for IoT devices in smart logistics based cyber-physical systems: bonded coverage and connectivity analysis. IEEE Access 2020;8:138350–72.
[27] Kshirsagar et al., (2022) [Kshirsagar, D., & Kumar, S. (2022). A feature reduction based reflected and exploited DDoS attacks detection system. Journal of Ambient Intelligence and Humanized Computing, 1-13.
[28] Pandey S, Singh RK, Gunasekaran A, Kaushik A. Cyber security risks in globalized supply chains: conceptual framework. Journal of Global Operations and Strategic Sourcing 2020;13(1):103–28.
[29] Saghezchi FB, Mantas G, Violas MA, de Oliveira Duarte AM, Rodriguez J. Machine learning for DDoS attack detection in industry 4.0 CPPSs. Electronics 2022;11(4):602.
[30] Cil AE, Yildiz K, Buldu A. Detection of DDoS attacks with feed forward based deep neural network model. Expert Syst Appl 2021;169:114520.
[31] Aldhyani TH, Alkahtani H. Cyber security for detecting distributed denial of service attacks in agriculture 4.0: deep learning model. Mathematics 2023;11(1):233.

[32] Mohammadian H, Ghorbani AA, Lashkari AH. A gradient-based approach for adversarial attack on deep learning-based network intrusion detection systems. Appl Soft Comput 2023;137:110173.
[33] Sambangi S, Gondi L. (2020, December). A machine learning approach for ddos (distributed denial of service) attack detection using multiple linear regression. In Proceedings (Vol. 63, No. 1, p. 51). MDPI.
[34] Le TT, Kim H, Kang H, Kim H. Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. Sensors 2022;22:1154.
[35] Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Futur Gener Comput Syst 2019;100:779–96.