Research Article

International Journal of Advanced Computer Research, Vol 13(65)
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2023.1362035

Design of advanced intrusion detection system for multimodal fuzzy-based convolutional neural networks (mFCNN)

K. Suresh Kumar[1], T. Ananth Kumar[2*] and R. Nishanth[3]
Assistant Professor, Information Technology, Sri Krishna College of Technology, Coimbatore, India[1]
Associate Professor, Computer Science and Engineering, IFET College of Engineering, Villupuram, Tamil Nadu, India[2]
Assistant Professor, Department of Electronics and Communication Engineering, Cochin University College of Engineering Kuttanad, Pullinkunnu, Kerala, India[3]

Received: 20-June-2023; Revised: 11-November-2023; Accepted: 15-November-2023

Abstract
The Internet of Things (IoT) develops a smart autonomous system capable of controlling and facilitating numerous human intervention activities such as inventory, electricity, traffic, and health management. It primarily operates by interconnecting networks and objects in various locations. Communication is carried out between devices to accomplish diverse missions in industrial, household, and healthcare applications. The involvement of the IoT in these applications renders the devices susceptible to cyber-attacks. In this work, an intelligent methodology was proposed to defend the devices against various security-related threats using modified deep learning algorithms. The intrusion detection system was designed using a hybrid model comprising multimodal fuzzy-based convolutional neural networks (mFCNN) in association with bidirectional long-short term memory (Bi-LSTM) and the enhanced code element embedding (ECEE) method. Here, the input for training is derived from real images, videos, and audio information retrieved from various sensors. Initially, the ECEE in each input frame is converted into suitable vectors. These vectors are then mapped to corresponding fixed-length vectors for embedding to achieve the most compressed representation. The Bi-LSTM is utilized for extracting relevant information related to spatial features, thus providing an effective intrusion detection mechanism. The mFCNN extracts the most critical temporal features by classifying contextual inputs using the captured videos and images. The experimental results demonstrate that the proposed hybrid model yields better results compared to existing methodologies, with the accuracy of protection benchmarked at 99.91%, which is higher than other baseline methods.

Keywords
Internet of Things (IoT), Cybersecurity, Deep learning algorithms, Intrusion detection systems, Hybrid neural networks.

1.Introduction
The Internet of Things (IoT) is a rapidly expanding global network that connects both living and nonliving entities. A security flaw within IoT renders it susceptible to a wide array of cyberattacks [1]. Users might be reluctant to embrace new technology if the network is perceived as insecure. While IoT is a remarkable concept, its global network of connected smart objects makes it particularly prone to cyberattacks [2]. Intrusion detection systems (IDS) can enhance the security of any network environment. Most modern IDSs employ machine learning (ML) algorithms to train and detect network attacks [3].

Despite their advancement, new IoT networks are vulnerable to a wide range of security threats and vulnerabilities due to the inefficient processing power of these devices [4]. Due to technological advancements, IDS are now extremely effective network security tools. There has been a significant increase in IoT device usage in recent years. Devices with IoT capabilities can connect to the internet and transmit data [5], accessible and controllable from any location due to the internet. IoT is a network connecting physical and digital devices, not just a computer network [6]. It has application domains in industry, home automation, healthcare, and other innovative technologies. ML, particularly deep learning (DL), is gaining popularity in various scientific fields, including IoT [7]. DL approaches are automatically used in IoT security for detecting network intrusions and malware. For example, the

*Author for correspondence

automatic classification of fish based on inherent characteristics is well-known [7]. Contrary to popular belief, DL-based IoT analysis can be easily circumvented; hostile environments are examined for malicious tasks [8]. Malware classifiers for PDF files can be deceived into marking malicious files as benign. Currently, DL techniques are not secure in IoT environments [9]. However, DL has far-reaching implications, which is intriguing. Information retrieval has gained public attention due to its wide range of applications [10]. While DL-based retrieval of images and text is well-studied, retrieving other formats like graphs is more challenging. The deep relevance matching model (DRMM) outperforms traditional models by addressing key relevance matching factors [11] and can be built on top of long-term memory. DL-based data filtering systems could become more effective as they start with a predefined measure of similarity derived from data measurements, but these metrics may not always be appropriate. DL can help analysts in resolving complex situations. IoT's significance continues to grow, especially for diverse and open mobile IoT-enabled health networks, which makes large amounts of healthcare data vulnerable [12]. Healthcare organizations may face network and security issues. Xu et al. explored cooperative network security using Weibull fading channels [13], and the 2-Rayleigh model accurately simulates vehicle-created propagation environments.

The objective of the paper is to enhance the stability and efficiency of intrusion detection by employing a hybrid model that combines multimodal fuzzy-based convolutional neural networks (mFCNN) with bidirectional long-short term memory (Bi-LSTM) and the enhanced code element embedding (ECEE) method. This approach aims to classify network traffic, extract critical features, and analyze the relationships between data packets to effectively detect and prevent a wide range of cyber threats. Additionally, the paper seeks to address the challenges in IoT security by proposing a robust, deep learning-based solution designed to handle complex and dynamic network environments.

The remaining text is structured as follows: Section 2 discusses previous research. Section 3 describes the datasets and pre-processing methods used to create them. Section 4 presents the evaluation results with a variety of hyperparameters and discusses the implications of our findings. Finally, it is concluded in Section 5.

2.Related works

The use of ML algorithms for attack detection was proposed by [14], highlighting the need for edge processing in large IoT networks due to the limitations of cloud computing. A novel approach using a support vector machine (SVM) and a self-taught DL autoencoder for intrusion detection was proposed by [15], which improves SVM classifier performance while saving time and costs. In [16] contrasts shallow and deep neural networks, suggesting DL's potential to enhance cybersecurity, with a study [17] showing a three-layer deep neural network model's superiority. However, the application of ML or DL in cybersecurity is challenging, with many zero-day attacks being discovered in IoT due to various protocols used by devices [18]. A new intrusion detection approach was developed, employing the intrusion label and decoder layer alongside common vulnerabilities and exposures (CVE) to prevent unauthorized access [18]. Another method [19] predicts security breaches using the Nash equilibrium. He et al. [20] studied vehicle networks using the two-Rayleigh model, while Feng et al. [21] developed analytical expressions for amplify and forward (AF) relaying networks.

In physical layer security, the 2-Rayleigh and Nakagami-m models are under scrutiny. Traditional communications use these models, but mobile IoT communication is more dynamic and complex [22]. The antenna for mobile vehicular communication moves constantly, resulting in multiple scattering groups, which classic fading channels like the Nakagami-m and 2-Rayleigh [23] do not account for. Tegos et al. [24] adapted the Nakagami model for mobile environments to assess the performance of mobile secrecy systems. Environmental changes must be promptly reported for secure device communication. The healthcare sector, particularly in IoT, has seen increased AI application [25]. Kumar et al. [26] employed general regression methods, Alhussein et al. [27] proposed an emotion prediction system from healthcare big data, Samuel et al. [28] suggested back-propagation networks for hospital inventory, and Cao et al. [29] discussed using convolutional neural networks (CNNs) for image processing. Khan et al. [30] compared ML techniques for intrusion detection, Alqahtani et al. [31] developed the GXGBoost model using genetic algorithms and the XGBoost classifier, and Derhab et al. [32] proposed a method to prevent command forgery using blockchain and software-defined networks (SDN). DL systems, which are widely used

in IDS [32], do not require manual feature extraction, increasing precision in threat detection. However, DL-based intrusion detection requires extensive training data, which is challenging due to the scarcity of attack traffic in real-world datasets like “KDD99”, “NSL-KDD”, and “CICIDS2017” and the difficulty in capturing and simulating some attack traffic, making it hard to assess specific attacks in detail using DL-based methods. While DL reduces overall training time with large datasets, it does not require prior feature selection for learning and testing classification rules.

The review indicates a focus on enhancing cybersecurity in IoT using ML and DL, despite challenges like zero-day attacks and the need for extensive training data. Novel approaches and models are continuously proposed to address these evolving security concerns.

3.Methods

3.1Multimodal fuzzy-based convolutional neural networks(mFCNN)

This section analyzed a novel method for defending devices against various security-related attacks. The mFCNN, an improved method for learning and classifying traffic packets considering both the temporal passage and physical location, was introduced. Consequently, class weight was utilized to enhance the model's stability. The overall architecture of the proposed model was depicted in *Figure 1*.

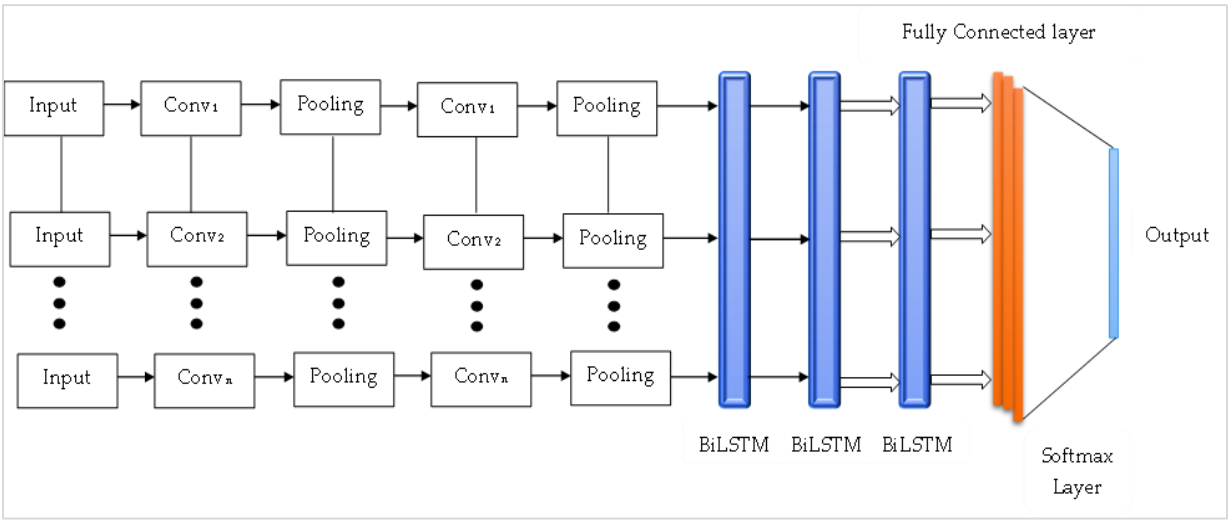

Figure 1 Architecture of proposed system

The classifier used here is the combination of mFCNN in association with Bi-LSTM and the ECEE method. The mFCNN part consists of the input layer and the enhanced code element embedding layer. Totally n number of convolution layer is used along with the pooling layer and fully connected layer. After the content of the information that has been preprocessed has been received, the mFCNN section will begin to process it, and the processing will result in the production of a high-dimensional vector representing the BiLSTM section. The BiLSTM section consists of an output layer, a SoftMax layer, and three layers of the BiLSTM algorithm itself and the output and SoftMax layers. A processing algorithm is applied to collect high-dimensional vectors in the proposed model. The algorithm then returns the results in the form of a probability of sequences that correspond to the system's membership in the various classes. The SoftMax layer takes the most stringent classification as its starting point and determines the final output from there. A new estimate of the probability is produced as an output of the process.

The data packets, initially converted into two-dimensional matrices, undergo preprocessing to generate images. These images result from the amalgamation of various bit values found in the respective network traffic locations. The electronic data and other crucial information in network traffic packets consist of bytes, the fundamental units of this data. These bytes are valued on a scale ranging between 0 and 255. This is just like the image bytes. Consider the byte in the packet header $\propto$ and payload

byte as β. These ∝ and β is combined with the traffic images for further processing. The mFCNN consists of input, convolution layers, pooling layer, fully connected layer, SoftMax layer, and the output layer. The combination of these layers allows the input layer from various aspects. Totally n number of sets are there in the setup. Of these, the first combination is used for extracting the local features of the traffic images. The exact features and the stability-filled results are obtained in the pooling layer. Then another set of combinations might yield a very large convolution kernel for analysing the similarities in between the two bytes that are not near which is same as that of the information spread in the traffic payload. After pre-processing, the network traffic might establish the vector input in the input layer. Then the input layer might set up with the data length that is intercepted from the corresponding i[th] packet i.e., $P_i = (\Gamma_1, \Gamma_2, \ldots, \Gamma_L)$ which is followed by the combination C of m packets related information set $IS = (P_1, P_2, \ldots P_n)$. The convolution layer with side length S_l is mathematically expressed as Equation 1.

$$V^{l+1}(a,b) = \sum_{a=1}^{n}\sum_{b=1}^{m}\left[{V_a}^1(s \times a + \alpha, s \times b + \beta) \oplus {R_k}^{l+1}(\alpha,\beta)\right] + \sum_{L-1}^{k} b(a,b) + \alpha\beta \quad (1)$$

In addition to that the value of (a,b) is obtained as Equation 2.

$$(a,b) \in \{0,1,2,3,\ldots M_{l+1}\} M_{l+1} = \left((M_l + 4P - S_l)/stride\right) + 1 \quad (2)$$

Where Ck is the plotted channel count in the characteristics graph, V denotes the output value in consideration with the activation function, the layer that is used for convolution may consist of an activation function that has an expression that has a lot of complex features due to the fact that V represents the output value in consideration with the activation function. The vector representing the output is shown in the Equation 3 as,

$${V_{a,b,k}}^1 = S_l\left({T_{a,b,k}}^i\right) + {R_x}^1(Tx) \quad (3)$$

The convolutional layer is the one that is responsible for the extraction of the features, and after the features have been extracted, the output image undergoes a complete transformation so that it can be sent to the max-pooling section, where a feature can be selected and filtered.

The pooling section consists of an already set in pooling function which can replace the outcome in a separate section corresponding to the adjacent region. The pooling layer is computed using the mathematical expression (Equation 4).

$${V_k}^l(a,b) = \left[\sum_{a=1}^{S_l} {V_k}^l(s \times a + \alpha)^R + \sum_{b=1}^{S_l} {V_k}^l(s \times b + \beta)^R\right]^{1/R} \quad (4)$$

3.2Feature extraction using Bi LSTM

Both regular network connections and network attacks are conducted using a specific network protocol. The packets that are affected by the attacks are encountered in the packets that are alongside the traffic. When considering the network protocol, it is contained in the fixed regions as the normal connections are established by the key exchanges incorporating the connections and the other disconnections. Normally, traffic attacks are used to determine the main cause of the attacks. This leads to data that is difficult to label, reveals an excessive amount of unwanted traffic, and ultimately produces poor training results when using a single mFCNN to train the properties of a single packet as the foundation of the algorithm. This is because the data is difficult to label and reveals an excessive amount of unwanted traffic. This is due to the fact that the algorithm relies on a single packet as its primary building block. Additionally, this exposes an excessive amount of unwanted traffic. This could be reduced by introducing the BiLSTM, which is intended to take the information containing the single connection or as a group that determines all the data packets in the mentioned groups and for determining the basic nature of either the traffic details or the packet information's. Natural language processing outperforms the processing scenario of the traffic or other network issues by using the same proposed methodology.

The BiLSTM region contains the various layers used for classification purpose. It contains n number of convolutional layers in addition to the other fully connected layers along with the SoftMax layers and the output layers. Two BiLSTM layers carry out the major functions. The BiLSTM is a unique neural networks recurrent neural networks (RNN) that was created to overcome gradients disappearing in addition with gradient explosion issues during extended sequential training. Generally, the recurrent RNN might have the tanh layer but the other BiLSTM networks will sequentially outperform the situation by predicting the better performance time in consideration of the memorable gates. Here the BiLSTM node of a cell is represented as Nc, then the input and output are expressed as x(t) and y(t). The initial step in the BiLSTM layer is to determine the model-related information that can discard the cell's state. The initial decision is made with the gate forgetting. The gate terminal reads the input and the

outputs by considering the values between 0 and 1. Each number in the Nc cell must contain a stable state by which '1' corresponds to the 'completely retained' and '0' corresponds to the 'completely discarded'. Wt represents the weights and the bias in the neural network is represented by β. It is mathematically expressed as Equation 5.

$$F_l = \sigma(\chi_f . [y_{t-a}, x_t] + \beta_f) \quad (5)$$

The other step is for deciding the range of novel information for adding the state of the cell. Initially the sigmoid layer is used for determining the needed information that needs to be modified using Equation 6.

$$I_t = \sigma(\chi_f . [y_{t-a}, x_t] + \beta_f) \quad (6)$$

The alternating vector for updating the security status is generated by using the tanh layer. This is mathematically expressed using the Equation 7.

$$N_c = tanh(W_t . [y_{t-1}, x_t] + \beta_c) \quad (7)$$

The two regions are then combined using the cell state by updating the formula (Equation 8).

$$N_c = F_t * N_{c-1} + i_t \times \bar{N}_c \quad (8)$$

The gate at the output might determine the output prediction of the cell (Equation 9).

$$Q_r = \sigma(\chi_f . [y_{t-a}, x_t] + \beta_r) \quad (9)$$

Then the cell state is processed through the tanh function for obtaining the formula (Equation 10).

$$Y_t = Q_r \times tanh(N_c) \quad (10)$$

The proposed model, which includes the feature maps, as each data packet containing a collection of traffic images generated in conjunction with the BiLSTM section. The relationships that exist between the features are structured in the spaces that exist between the 'd' data packets, which are then analysed using the BiLSTM layers. These initial few packets are also used. The following available packets could have very long payloads, which would include the data related to the attack.

The proposed system's BiLSTM layer may incorporate an activation function designed to reduce training time. Typically, the rectified linear unit (ReLU) function triggers the BiLSTM's activation. The standard process involves a system used for multi-classification, where the model is trained using multi-class cross-entropy.

3.3Class weights

The data obtained after the pre-processing mode is shown in *Table 1*.

Table 1 Dataset related metrices

Label	Types of attacks	Number
1	Physical Attacks	67432
2	Encryption Attacks	45662
3	Denial of Service (DoS)	456
4	Firmware Hijacking	6532
5	Botnets	7654
6	Man-in-the-Middle	9823
7	Ransomware	4563
8	Eavesdropping	784
9	Privilege Escalation	8943
10	Brute Force Password Attack	9045

Here the numbers of various types of data are selected unevenly. When compared to the other types, the number of type '0' is the highest, while the other types are ranked as the lowest. This may affect the outcome of the classification section. The class weight is calculated using the Equation 11.

$$W_i = \frac{\sum_{i=0}^{N-1} \eta_i}{\eta_i} \quad (11)$$

4.Results and discussion

The dataset utilized for the experimentation was sourced from the network intrusion detection section on the https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection website. It comprises specific parameters relevant to the study. It normally specifies the influence of the size of data packets in the process of training, the impact of the number of packets per flow, and the effect of the sample size chosen in addition to the total effect of the units used for representing the BiLSTM by influencing the weight classes. The proposed model-related parameters are optimized in such a way that the mFCNN is compared with the other neural network models along with the BiLSTM. The ratio of the datasets used for training, testing, and validation is made in the ratio of 3:6:16. The parameters used for the analysis are the accuracy, precision, recall, sensitivity, and the f score value. The overall performance of the proposed model is specified in terms of accuracy, the precision represents the ratio of the positive samples in consideration with the total samples and the negative sample consideration with the total samples is specified in terms of sensitivity. The recall values specify the number of true positive samples divided in terms of true positives and false negatives. The classifier's accuracy is defined with

the harmonic weights which specifies the F score value.

4.1Experimentation

The experiment was conducted on a Linux operating system using an Intel Core i5 processor, equipped with a 256 GB SSD. The setup included Anaconda 4.5.10, Keras 2.2.20, and Python 3.9.

4.2Performance of the model based on BiLSTM

In the framework of the BiLSTM, the value of the system's output parameter is modelled as a series of units. During the research, it was concluded that increasing the number of BiLSTM units initially improves the model's performance before it begins to deteriorate. After careful consideration, the optimal number of BiLSTM units was determined to be 121.

4.3Impact of Model performance in specifying the packet length of the training packets

The training results indicate an increase in the model's quality, which starts to decline when the package length exceeds 70. This is likely because retraining with excessively long data packets increases the proportion of smaller packets, thereby raising the fraction of units with longer lengths and reducing the model's accuracy. The median value of 1 further lowers the calculation model's precision.

The study considers various attacks, including Physical Attacks, Encryption Attacks, DoS, Firmware Hijacking, Botnets, Man-in-the-Middle, Ransomware, Eavesdropping, Privilege Escalation, and Brute Force Password attacks. *Table 2* presents these types of attacks along with their corresponding values, considering specified parameters like accuracy, precision, recall, and f-score.

It reflects the predicted changes in values relative to the increase in packet length during training. When this length exceeds a maximum threshold, the effectiveness of the data packets is ascertained by evaluating their scientific credibility incorporated into the training sectors. *Figure 2* demonstrates the impact on model performance as processed with varying training packet lengths.

Table 2 Values for various types of attacks

Types of attacks	Accuracy	Precision	Recall	Sensitivity	F Score
Physical Attacks	99.97	99.31	0.12	90.99	98.85
Encryption Attacks	99.98	99.32	0.14	91	98.86
DoS	99.96	99.3	0.31	90.98	98.84
Firmware Hijacking	100	99.34	0.22	91.02	98.88
Botnets	99.68	99.02	0.62	90.7	98.56
Man-in-the-Middle	99.78	99.12	0.01	90.8	98.66
Ransomware	99.98	99.32	0.05	91	98.86
Eavesdropping	99.91	99.25	0.08	90.93	98.79
Privilege Escalation	99.92	99.26	0.21	90.94	98.8
Brute Force Password Attack	99.93	99.27	0.36	90.95	98.81
Overall	99.911	99.251	0.212	90.931	98.791

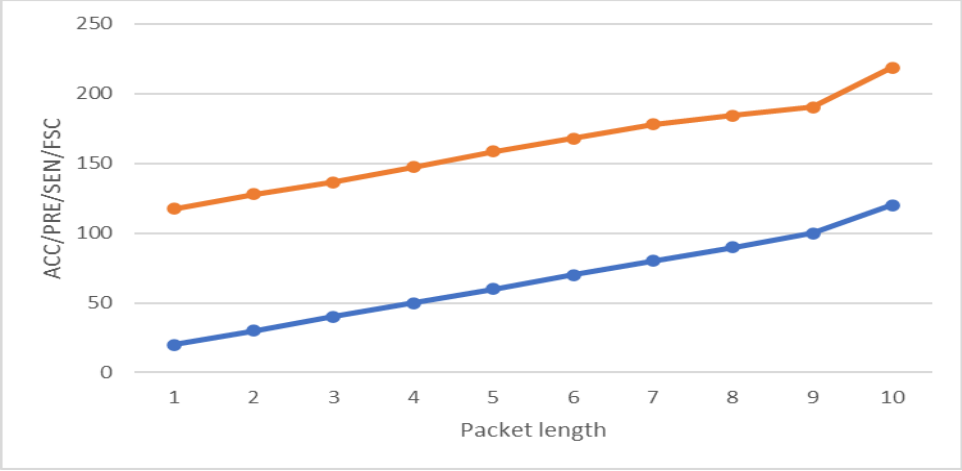

Figure 2 Determining the Proposed model performance with respect to packet length

This also mitigates the effects of overfitting, thereby ensuring the most effective accuracy setting for the classification and determining the accuracy in identifying data packets along with their packet headers.

4.4Performance of the proposed model in the determination of the packet flow quantity

There exists a possibility that increasing the specified number of data packets in each flow involved in the training process enhances the specificity of feature extraction. However, setting this value too high may lead to a rise in the percentage of filled data packets. An increase in the specified number of data packets in each flow is thought to make the feature extraction more specific. Nevertheless, if this value is excessively high, the percentage of already filled data packets increases, consequently decreasing the model's ability to extract information. The influence of the total number of packets in a flow on the results of a simulation, as demonstrated in *Figure 3*, shows that the model's performance significantly deteriorates when the number of packets per flow exceeds 12.

Performance of the model in the effect of the batch size

Batch size is an important criterion to consider during training models. Increasing the batch size while staying within a reasonable range can help improve memory utilization and lead to a speedier processing of data. However, if it is raised to an excessive range, it might severely slow down the process. The total size of the batch is proven to be the performance effect in the proposed models, as shown in *Figure 4*.

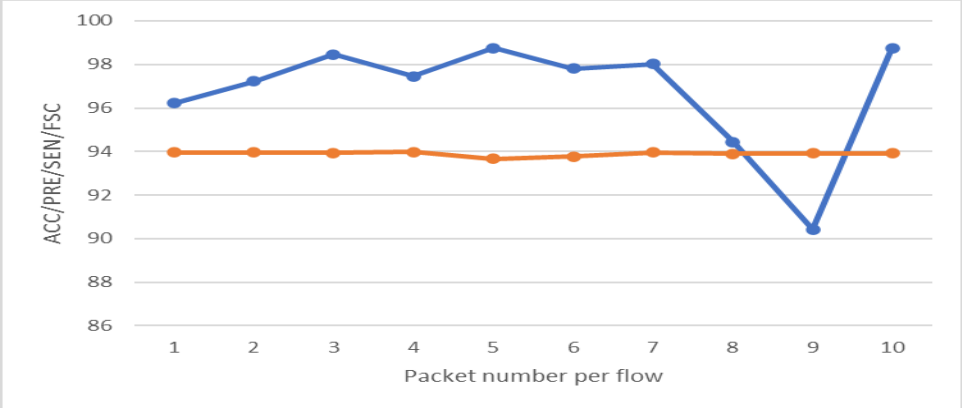

Figure 3 Proposed model performance with respect to packet per-flow

Figure 4 Performance of the model in the effect of the batch size

4.5Performance model for determining the effects of class weights

In *Table 3*, the results of two distinct groups of experiments, one with class weights and one without, are compared. The introduction of the class weight appears to lessen the influence of the existing dataset's unbalance in the quantity of data from various categories on simulation results. *Figure 5* illustrates the performance model for determining the effects of class weights in consideration of the available dataset in determining the quantity of the data.

Table 3 Performance model for determining the effects of Class Weights

Parameters	With class weights (in%)	Without class weights (in%)
Accuracy	99.25	97.45
Precision	96.35	96.25
Recall	0.87	0.45
Sensitivity	94.25	95.87
F Score	97.12	96.32

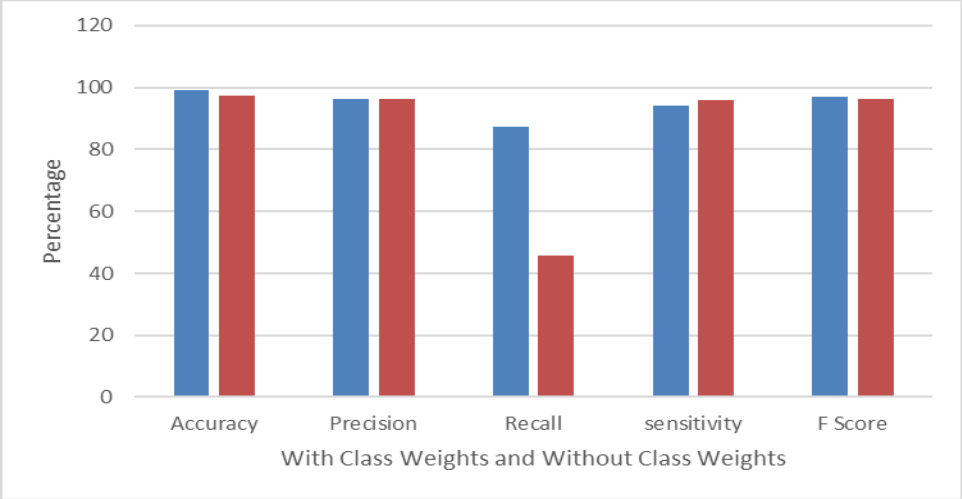

Figure 5 Performance model for determining the effects of class weights

The BiLSTM is used effectively for extracting the sequential connection between the packets. *Table 4* compares accuracy and other parameters obtained between the proposed models and other models in consideration of the types of attacks. The BiLSTM appears to be improved effectively to prove the efficiency of identification in consideration of the parameters by specifying the time duration of the attacks. When compared to BiLSTM model, the addition of the mFCNN will improve further for efficiency identification with more traffic attacks. The proposed mFCNN in combination with the BiLSTM detects the intrusion more effectively with a very less alarm rate and thus can be classified by setting the parameters of network traffic more accurately when compared to the mFCNN or BiLSTM. The comparative results prove that the proposed model provides better performance when compared to the other existing methods. A complete list of abbreviations is summarised in *Appendix I.*

Table 4 Comparison of the proposed model with other models

Methods	Accuracy	Precision	Recall	Sensitivity	F Score
CNN	94.65	95.64	0.53	92.78	94.36
K-nearest neighbor algorithm (KNN)	93.65	94.64	0.62	91.78	93.36
LSTM	94.63	95.62	0.34	92.76	94.34
RNN	92.56	93.55	0.85	90.69	92.27
SVM	91.35	92.34	0.62	89.48	91.06
CNN+LSTM	96.62	97.61	0.12	94.75	96.33
RNN+LSTM	95.78	96.77	0.64	93.91	95.49
Proposed	99.911	99.251	0.212	90.931	98.791

5.Conclusion

DL algorithms address security threats in IoT environments. In this work, an intelligent methodology was proposed to defend devices against various security threats using modified DL algorithms. The intrusion detection system is designed using a hybrid model comprising mFCNN in association with Bi-LSTM and the ECEE method. Bi-LSTM is utilized to extract relevant spatial feature information, providing an effective intrusion detection mechanism. Meanwhile, mFCNN classifies contextual inputs, including captured videos and images, to extract essential temporal features. The

experimental results demonstrate that the proposed hybrid model outperforms existing methodologies, achieving a benchmarked accuracy of 99.91% for protection, which is higher compared to other baseline methods.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

References
[1] Kuzlu M, Fair C, Guler O. Role of artificial intelligence in the internet of things (IoT) cybersecurity. Discover Internet of Things. 2021; 1:1-4.
[2] Rahman SA, Tout H, Talhi C, Mourad A. Internet of things intrusion detection: Centralized, on-device, or federated learning? IEEE Network. 2020; 34(6):310-7.
[3] Dang QV. Studying machine learning techniques for intrusion detection systems. In future data and security engineering: 6th international conference, Nha Trang City, Vietnam, proceedings,2019 (pp. 411-26). Springer International Publishing.
[4] Rajmohan R, Kumar TA, Julie EG, Robinson YH, Vimal S, Kadry S, et al. G-Sep:a deep learning algorithm for detection of long-term sepsis using bidirectional gated recurrent unit. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 2022; 30(Supp01):1-29.
[5] Laghari AA, Wu K, Laghari RA, Ali M, Khan AA. A review and state of art of Internet of Things (IoT). Archives of Computational Methods in Engineering. 2021: 1-9.
[6] Kumar KS, Mani AR, Sundaresan S, Kumar TA, Robinson YH. Blockchain-based energy-efficient smart green city in IoT environments. In blockchain for smart cities 2021 (pp. 81-103). Elsevier.
[7] Magán-Carrión R, Urda D, Díaz-Cano I, Dorronsoro B. Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. Applied Sciences. 2020; 10(5):1775.
[8] Kumar A, Purohit V, Bharti V, Singh R, Singh SK. Medisecfed: Private and secure medical image classification in the presence of malicious clients. IEEE Transactions on Industrial Informatics. 2021; 18(8):5648-57.
[9] Rajmohan R, Pavithra M, Kumar TA, Manjubala P. Exploration of deep RNN architectures: LSTM and gru in medical diagnostics of cardiovascular and neuro diseases. In handbook of deep learning in biomedical engineering and health informatics 2021 (pp. 167-202). Apple Academic Press.
[10] Sundaresan S, Kumar KS, Kumar TA, Ashok V, Julie EG. Blockchain architecture for intelligent water management system in smart cities. In blockchain for smart cities 2021 (pp. 57-80). Elsevier.
[11] Hao S, Shi C, Cao L, Niu Z, Guo P. Learning deep relevance couplings for ad-hoc document retrieval. Expert Systems with Applications. 2021; 183:115335.
[12] Otoum Y, Liu D, Nayak A. DL-IDS: a deep learning–based intrusion detection framework for securing IoT. Transactions on Emerging Telecommunications Technologies. 2022; 33(3):e3803.
[13] Xu L, Zhou X, Tao Y, Liu L, Yu X, Kumar N. Intelligent security performance prediction for IoT-enabled healthcare networks using an improved CNN. IEEE Transactions on Industrial Informatics. 2021; 18(3):2063-74.
[14] Elsayed MS, Le-Khac NA, Dev S, Jurcut AD. Machine-learning techniques for detecting attacks in SDN. In 7th international conference on computer science and network technology 2019 (pp. 277-81). IEEE.
[15] Binbusayyis A, Vaiyapuri T. Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM. Applied Intelligence. 2021; 51(10):7094-108.
[16] Kumar TA, Rajmohan R, Adithya M, Sunder R. A novel security scheme using deep learning based low overhead localised flooding algorithm for wireless sensor networks. International Journal of Data Science. 2021; 6(1):19-32.
[17] Sharma R, Vashisht V, Singh U. Performance analysis of evolutionary technique based partitional clustering algorithms for wireless sensor networks. In soft computing: theories and applications: proceedings of SoCTA 2018 (pp. 171-80). Springer Singapore.
[18] Pontes CF, De Souza MM, Gondim JJ, Bishop M, Marotta MA. A new method for flow-based network intrusion detection using the inverse Potts model. IEEE Transactions on Network and Service Management. 2021; 18(2):1125-36.
[19] Martins I, Resende JS, Sousa PR, Silva S, Antunes L, Gama J. Host-based IDS: a review and open issues of an anomaly detection system in IoT. Future Generation Computer Systems. 2022; 133:95-113.
[20] He R, Schneider C, Ai B, Wang G, Zhong Z, Dupleich DA, et al. Propagation channels of 5G millimeter-wave vehicle-to-vehicle communications: recent advances and future challenges. IEEE Vehicular Technology Magazine. 2019; 15(1):16-26.
[21] Feng YH, Dai YJ, Wang RZ, Ge TS. Insights into desiccant-based internally-cooled dehumidification using porous sorbents: From a modeling viewpoint. Applied Energy. 2022; 311:118732.
[22] Zhou Z, Wang J, Ye Y. Exact BER analysis of differential chaos shift keying communication system in fading channels. Wireless Personal Communications. 2010; 53:299-310.
[23] Angjelichinoski M, Trillingsgaard KF, Popovski P. A statistical learning approach to ultra-reliable low latency communication. IEEE Transactions on Communications. 2019; 67(7):5153-66.
[24] Tegos SA, Diamantoulakis PD, Karagiannidis GK. On the performance of uplink rate-splitting multiple

access. IEEE Communications Letters. 2022; 26(3):523-7.
[25] Arumugam D, Govindaraju K, Tamilarasan AK. AIIoT-based smart framework for screening specific learning disabilities. In machine learning for critical internet of medical things: applications and use cases 2022 (pp. 103-24). Cham: Springer International Publishing.
[26] Kumar TA, John A, Kumar CR. 2. IoT technology and applications. Internet of Things. 2020:43-62.
[27] Alhussein M, Muhammad G, Hossain MS, Amin SU. Cognitive IoT-cloud integration for smart healthcare: case study for epileptic seizure detection and monitoring. Mobile Networks and Applications. 2018; 23:1624-35.
[28] Samuel TA, Pavithra M, Mohan RR. LIFI-based radiation-free monitoring and transmission device for hospitals/public places. In multimedia and sensory input for augmented, mixed, and virtual reality 2021 (pp. 195-205). IGI Global.
[29] Cao C, Liu X, Yang Y, Yu Y, Wang J, Wang Z, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In proceedings of the IEEE international conference on computer vision 2015 (pp. 2956-64).
[30] Khan FA, Gumaei A, Derhab A, Hussain A. A novel two-stage deep learning model for efficient network intrusion detection. IEEE Access. 2019; 7:30373-85.
[31] Alqahtani M, Gumaei A, Mathkour H, Maher Ben Ismail M. A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks. Sensors. 2019; 19(20):4383.
[32] Derhab A, Guerroumi M, Gumaei A, Maglaras L, Ferrag MA, Mukherjee M, et al. Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security. Sensors. 2019; 19(14):3119.

K. Suresh Kumar is currently pursuing his Ph.D. in Information and Communication Engineering at Anna University, Chennai. He obtained his Master's degree in Computer and Information Technology from Manonmaniam Sundaranar University, Tirunelveli, during 2009-2011, and his Bachelor's degree in Electronics and Communication Engineering from Anna University, Chennai, during 2004-2008. Presently, he serves as an Assistant Professor at Sri Krishna College of Technology, affiliated with Anna University, Chennai. He has contributed papers to various national and international conferences and journals. His areas of interest include Computer Networks, Data Analytics, and Natural Language Processing. Additionally, he is a life member of ISTE and holds memberships in several other bodies.
Email: sureshkumar.k@skct.edu.in

T. Ananth Kumar received his Bachelor's degree in Electronics and Communication Engineering from Anna University, Chennai, and his Master's degree in VLSI Design from Easwari Engineering College, affiliated with Anna University. He earned his Ph.D. in VLSI Design from Manonmaniam Sundaranar University, Tirunelveli. He has since become an Associate Professor at IFET College of Engineering, India. He has presented papers at various national and international conferences and journals. His fields of interest include Networks on Chips, Computer Architecture, and ASIC design. He is a recipient of the Best Paper Award at INCODS 2017 and holds several patents.
Email: tananthkumar@ifet.ac.in

R. Nishanth serves as an Assistant Professor in the Electronics and Communication Engineering Department at Cochin University College of Engineering Kuttanad, India. He obtained his Bachelor's degree in Electronics and Communication Engineering and his Master's degree in Applied Electronics from Anna University, Chennai, Tamilnadu. Currently, he is pursuing his Ph.D. in Information and Communication Engineering from Anna University, Chennai. With several years of experience in teaching and research, he has organized and participated in numerous conferences, seminars, workshops, and various other events. His research interests lie in VLSI Design and Image Processing.

Appendix I

S. No.	Abbreviation	Description
1	AF	Amplify and Forward
2	AI	Artificial Intelligence
3	Bi-LSTM	Bidirectional Long-Short Term Memory
4	CNN	Convolutional Neural Networks
5	CVE	Common Vulnerabilities and Exposures
6	DL	Deep Learning
7	DoS	Denial of Service
8	DRMM	Deep Relevance Matching Model
9	ECEE	Enhanced Code Element Embedding
10	IDS	Intrusion Detection Systems
11	IoT	Internet of Things
12	KNN	K-Nearest Neighbor Algorithm
13	LSTM	Long-Short Term Memory
14	mFCNN	Multimodal Fuzzy-Based Convolutional Neural Networks
15	ML	Machine Learning
16	PDF	Portable Document Format
17	ReLU	Rectified Linear Unit
18	RNN	Recurrent Neural Networks
19	SDN	Software-Defined Networks
20	SVM	Support Vector Machine