Information Dynamics and Applications
https://www.acadlore.com/journals/IDA

Routing Attack Detection Using Ensemble Deep Learning Model for IIoT

Ramesh Vatambeti[1*], Gowtham Mamidisetti[2]

[1] School of Computer Science and Engineering, VIT-AP University, 522241 Vijayawada, India
[2] Department of AI & ML, Malla Reddy Institute of Technology & Science, 500100 Hyderabad, India

Correspondence: Ramesh Vatambeti (ramesh.v@vitap.ac.in)

Received: 01-21-2023 **Revised:** 02-28-2023 **Accepted:** 03-06-2023

Citation: R. Vatambeti and G. Mamidisetti, "Routing attack detection using ensemble deep learning model for IIoT," *Inf. Dyn. Appl.*, vol. 2, no. 1, pp. 31-41, 2023. https://doi.org/10.56578/ida020104.

Abstract: Smart cities, ITS, supply chains, and smart industries may all be developed with minimal human interaction thanks to the increasing prevalence of automation enabled by machine-type communication (MTC). Yet, MTC has substantial security difficulties because of diverse data, public network access, and an insufficient security mechanism. In this study, we develop a novel IIOT attack detection basis by joining the following four main steps: (a) data collection, (b) pre-processing, (c) attack detection, and (d) optimisation for high classification accuracy. At the initial stage of processing, known as "pre-processing," the collected raw data (input) is normalised. Attack detection requires the creation of an intelligent security architecture for IIoT networks. In this work, we present a learning model that can recognise previously unrecognised attacks on an IIoT network without the use of a labelled training set. An IoT network intrusion detection system-generated labelled dataset. The study also introduces a hybrid optimisation algorithm for pinpointing the optimal LSTM weight when it comes to intrusion detection. When trained on the labelled dataset provided by the proposed method, the improved LSTM outperforms the other models with a finding accuracy of 95%, as exposed in the research.

Keywords: Industrial IoT; Long Short-Term Memory; Attack Detection; Network connectivity; Unknown attacks

1. Introduction

The IIoT is a subset of the IoT that makes it possible to connect devices in a smart way to provide predictive services in an industrial sector that is becoming more and more automated [1]. Machine-type communication (MTC) is an example of pervasive communication that is needed for the interconnection of devices so that machines can collaborate on an IIoT task without human intervention. One machine gathers sensitive information from the business setting and sends it to others using a wireless or cellular network interface [2, 3]. The information is then analysed by a computer model, which makes precise judgements and perhaps initiates robotic processes. The Internet is used for machine-to-machine communication; however, this exposes MTC systems to a wide variety of security threats, with, but not incomplete to, attacks, network exploitation, injection attacks [4, 5]. Furthermore, the MTC system's many devices collect an enormous variety of data (industrial-critical data) that needs regular monitoring to prevent data breaches and tampering.

Due to variables such as the devices' limited resources, the network's dynamic topology, and the variety of attack vectors, detecting assaults on the routing scheme of IIoT strategies can be challenging [6]. Recent years have seen the development of a number of methods for dealing with this problem, machine learning-based approaches. Different aspects of network traffic, like traffic patterns, are used by these techniques to detect and categorise routing attacks [7, 8]. Particular attention is paid in this essay to the RPL protocol, which is vulnerable to routing attacks. Attacks against RPL may be broken down into two groups: those that are carried over from WSNs and those that are special to RPL and take advantage of its particular weaknesses [9]. This page details a variety of RPL assaults, such as Flood Attacks, Data-DoS/DDoS Attacks [10], which mostly target layer 3 of the OSI perfect.

In reality, the application layer is the highest level of the IIoT architecture [11], and it enables a wide variety of industrial processes and applications with smart healthcare, smart vehicles, and so on. The (IIoT) is an all-encompassing network that serves a wide range of industries and individual users. But, it brings up a wide range

of new issues relating to safety, security, the economy, and society. To address these issues, we need scalable solutions on a grand scale. Due to their limited resources, IoT sensor nodes necessitate security products that utilise as little space, power, and money as possible. These fixes should work with the industry's standard in communication protocols [12]. As IoT devices generate vast volumes of data across industrial applications, an IIoT system is enticing to cybercriminals [13]. The sheer volume of data suggests that traditional methods of data processing are inadequate for IoT and IIoT use cases. Thus, machine learning is one of the best computer models for incorporating IoT-device intelligence (ML).

Maintaining proper command of IIoT's massive industrial systems is a challenging endeavour. The ability to swiftly and safely understand and analyse vast volumes of data is crucial for computing systems in the modern day [14]. In addition, the latency and reliability of data transmission required high system capability and throughput. The overall performance of the industrial sector has been vastly improved thanks to "Deep learning (DL) algorithms" and models in terms of dependability and reliability. These algorithms show a lot of promise for fixing security issues in IIoT [15]. Unfortunately, they lack the necessary accuracy and have a higher computational cost. In order to provide a potential answer to attack detection, optimisation methods might be used in the deep learning model.

This paper's contributions are summed up as:

❖ The perfect must be able to unearth concealed patterns in classify network traffic as either malicious or benign in order to detect novel or previously undisclosed threats. We've employed a suite of clustering methods to get here. The results of many clustering algorithms are pooled together using a weighted voting approach to improve the accuracy with which the class label (malicious/non-malicious) is predicted for a given piece of IIoT network data. After conducting a thorough performance investigation, weights have been determined for the results of each clustering method. An unsupervised mechanism capable of identifying voting, which transforms an unlabelled dataset into a labelled dataset.

❖ A deep learning model for IoT network attack detection is trained using the labelled dataset produced by the proposed technique. The performance of several deep learning models (optimised LSTM, MLP, and DBN) has been compared to determine the most effective model for detecting threats in an Internet of Things (IoT) network.

❖ Hybrid optimisation model is used to choose the LSTM's weight appropriately.

The remaining sections of the paper are as shadows: In Part 2, we outline the current research on how to spot attacks in IIoT networks. The suggested model is described in depth in Section 3. Section 4 discusses the suggested model's implementation and the deep learning models that were employed. Section 5 wraps up the report and discusses where the research may go from here.

2. Related Works

A new MANET routing protocol based on reinforcement learning and named reputation opportunistic routing by Ryu and Kim [16] is proposed (RORQ). This protocol uses game theory to identify and blacklist rogue nodes in a network, allowing for more streamlined traffic flow. So, our approach can more efficiently locate a routing path in a hostile network. The simulation results demonstrated that the suggested technique outperformed other cutting-edge routing protocols. Gains 82% in average end-to-end delay, and up to 28% in energy efficiency were shown by the proposed method over other algorithms in the blackhole to 12% in energy competence were shown by the proposed method over other algorithms in the grayhole attack scenario.

To aid in the finding of jamming attacks, Obeidat et al. [17] developed a model to analyse the operation of VANETs while under jamming attacks, and they offer EVA (Enhancement Voting Algorithm) based on global Trust are exchanged. Route Error (RERR) and HELLO packets are utilised during the period of route maintenance. Because of their crucial role in routing, these packets are also appraised as part of the trust score. Although while misbehaving nodes are technically capable of processing these packets, they are less likely to be utilised than their well-behaving counterparts. The calculated global trust value is used to define three trust levels that will be used to determine the optimal routing decision in the NS3 simulation. Bonnmotion is used to design and analyse mobility scenarios, which are then used to probe the properties of mobile multi-hop networks. In order to develop a mechanism strategy, the scenarios were spread to the NS3 network simulators. In order to determine how well a network performs when subjected to jamming assaults, it is first analysed using a variety of quality-of-service (QoS) metrics and throughput (PDR) measures.

WSNs are a cornerstone of the (IoT), and Rabhi et al. [18] highlight their susceptibility to routing assaults in their presentation of the Routing power (RPL). We also offer a method for identifying three distinct forms of assault against RPL, and we highlight some recent research suggestions for doing so. We simulate four network scenarios using Contiki-Cooja, one benign and three malicious presenting different phase, where we employed WEKA, to determine whether the behaviour was benign or malicious according to the database. In this stage, we employ many distinct classification procedures, which collectively allow us to achieve a precision value greater than 96%.

To identify DDoS bouts in the IoT-CIDDS dataset, Malik et al. [19] suggest a feature engineering and machine learning outline. There are two stages to the framework: Our initial step is to create algorithms for dataset enrichment, with a focus on using cutting-edge feature engineering for statistical analysis of the dataset's probability distribution and feature correlations. Later, using IoT-CIDDS to generate training, validation, and testing datasets, we propose an ML model and conduct a complexity analysis of the feature-engineered dataset using five machine learning techniques. Performance metrics for training classifiers and evaluating ML models include false positive rate, accuracy, precision, recall, area under curve, and computational time. Detecting DDoS attacks in standard IoT networks using the 6LoWPAN stack is a challenging problem, but the experimental consequences show that significant feature reduction optimises the IDS.

Based on the DL technology, Alghamdi and Bellaiche [20] describe a cascaded wormhole detection method for Internet of Things networks (DTF). Using a federated strategy that ensures data security and privacy at the node level, (LSTM) deep learning models were trained. The DTF is based on two trust qualities. Due to its lightweight and accurate cascaded and federated learning strategy, the suggested method has achieved an accuracy of 96%.

Örs and Levi [21] offer a multi-class classifier based on machine learning that can distinguish between six different kinds of attacks and normal traffic. Instead of just having a general idea of whether or not attacks are happening on a network, our node-based feature extraction and detection approach models the traffic patterns of the attackers across a sliding time window, allowing us to pinpoint their exact IP addresses. We also present an intrusion detection dataset built from traffic data obtained from real-world IoT devices running 6LoWPAN and RPL protocols, which can be used for training and testing our algorithms. In addition to using RPL routing assaults, a common method of attack against IoT devices, we also make use of the Mirai botnet. As can be seen from the findings, the suggested intrusion detection system has a recall score between 79% and 100% for detecting 6 distinct types of attacks. We also deploy the generated model in an implementation across a testbed to demonstrate its viability.

3. Proposed System

3.1 Dataset Explanation

The LSTM-based discovery is trained with the X-IIOTID: connectivity and device-agnostic intrusion dataset for the IIoT [22] dataset. The final version of the dataset has a feature space size of 68 and contains 820834 training examples. There are three different kinds of attack labels that can be applied to a target: normal and attack, normal and sub-category attack, and normal and sub-sub-category attack. The algorithmic processing pipeline for the intelligence layer is depicted in Algorithm 1.

Algorithm 1. Data pre-processing algorithmic movement

Input: Raw machine requests D, Target Values $y_{m,1}$
Output: $\Phi_{m\times o\times n, y_{m,C}}$

$$y_{m,C} \leftarrow OHE(ym; 1)$$
$D \leftarrow drop(D, columns = [IP, date, timestamps, ids])$
if $D_{columns}$.isNull() and $D_{columns}$.dataType in [string,int] then
$D_{columns} \leftarrow D_{columns}.fillNull(D_{columns}.mode())$
else
$D_{columns} \leftarrow D_{columns}.fillNull(D_{columns}.median())$
end if
if $D_{columns}$.data Category is string then
$D_{columns}$ = labelEncoder ($D_{columns}$)
end if
$$\varphi m \times n \leftarrow D$$
$$\varphi_l \leftarrow groupBy(l)$$
$$\Phi_{m\times o\times n} \leftarrow \bigcup_{C} \phi_l$$
return $\Phi_{m\times o\times n}, y_{m,C}$

3.2 Preparing Datasets

In order to use the obtained dataset for training and prediction, it must first undergo pre-processing. Columns like IP addresses, dates, and ids that aren't strictly necessary are taken out of the dataset. All NaN and null values are replaced with the median of the related columns [23]. Columns containing strings can have their values converted to numbers using label encoding. Take the dataset represented by φ_(mn) where columns have been removed and null values have been substituted. Each class label l is transformed using the group by operation G.

$$\phi_l = G_l(\phi_{Instances \times features}) \forall l \in C \tag{1}$$

where, C represents a group of unique intended audiences. Moreover, it can be partitioned into a large number of timesteps, each of size o. For the last batch of training data, mon may be expressed as mon=C l. The string data type in each target class must be converted into a one-hot encoded vector. The dataset has C distinct classes, one of which is the initial target class, y. It's unmistakable that y is a string data type and that its form is (m,1). Changing y with a single pass of OHE,

$$y_{m,C} \coloneqq OHE(y_{m,1}) \tag{2}$$

This stands for the assignment operator. The following is one representation for the hot encoded vector y (m,C):

$$y_{m,C} = \begin{matrix} 0 & 0 \\ 0 & 0 \\ \vdots & \cdots \ \vdots \\ \lfloor 1 \rfloor & \lfloor 0 \rfloor \end{matrix} \quad \begin{matrix} \\ \\ \\ 0 \\ 1 \end{matrix} \tag{3}$$

3.3 Data Preparation

It is necessary to do feature engineering on the ensemble model to guarantee that the data is of the appropriate distribution before it can be functional to machine requests. Imagine a sudden influx of machine-generated queries D.

$$D = \{D_{M_1}, D_{M_2}, D_{M_3}, .., D_{M_i}, \dots, D_{M_m}\} \tag{4}$$

$$D_{M_i} = \{a_1, a_2, a_3, \dots, a_j, \dots., a_n\} \tag{5}$$

$$\forall 1 \leq i \leq m, \forall 1 \leq j \leq n$$

where, D_{Mi} is the data being communicated by the machine, and a_j is a single feature or characteristic of that data, and *n* is the total number of features. For at unit time period, active computers can create a request to send to the terminus node. The sending devices will provide you with these queries in the format of

$$\tau_t = \{t_1, t_2, \dots, t_k, \dots, t_o\} \tag{6}$$

$$\forall 1 \leq k \leq o, \ \tau_t \subseteq D$$

Following is a description of the form of the feature spaces for the associated machine data.

$$D_{M_i}, t_k = 1 \times n \tag{7}$$

$$\tau_t = \{(1 \times n), (1 \times n), \dots\}_{1 \times 0} \tag{8}$$

The size of the set t may be calculated as o_n. In order to acquire the final dataset fit for model predictions, τ_t is transformed to 1. The second dimension of the modified dataset τ_t represents the whole-time step, while the third dimension represents a number of characteristics. The model's specifications for the input shape inform the implementation of data preparation. It's important to remember that the size of the dataset being transmitted, τ_t, does not change.

3.4 Ensemble Learning Model with DL

The idea behind ensemble learning is that better performance may be achieved by combining the outcomes of many learning models. Independent ensemble building and coordinated ensemble construction are two implementations of the ensemble learning paradigm that can provide numerous projected outputs. The goal of the independent ensemble construction approach is to generate multiple results that can be joint using the ensemble technique by independently executing a learning algorithm multiple times on different training data subsets or by independently executing different learning models on the same dataset. In contrast, when building a coordinated ensemble.

The suggested model makes use of weighted voting to combine the results of many base learning models in an independent ensemble creation method. To predict a class label for each data vector in the given unlabelled dataset is the primary goal of the proposed ensemble learning model. As a result, we have relied on clustering methods to foretell the labels assigned to data matrices. Small Batch K-Means, Fuzzy C-Means, and OPTICS clustering were employed as the foundational learning models for the suggested model. Each clustering method will produce a 0 or 1 as its predicted output, with 0 representing benign traffic and 1 representing malicious traffic. Two groups, one containing benign data and the other containing harmful data, are created by combining the anticipated output from each clustering method, for each data entry, using weighted voting using equation 3.

Using a weighted voting system, the findings of many clustering algorithms are combined to create a single, more accurate forecast for the data. This is called an independent ensemble construction approach,

$$V = \sum_{i=1}^{3} (P_i * W_i) \tag{9}$$

where, W_i stands for the weights connected to the clustering method's base prediction P_i. Eventually, we get to the formula for predicting the class label V:

$$\hat{V} = \begin{cases} 1 & if\ V > 0.5 \\ 0 & otherwise \end{cases} \tag{10}$$

The suggested model's clustering approach, Small Batch K-Means, had its weights linked with the projected value adjusted to 0.25 for both OPTICS and Fuzzy C-Means after extensive performance investigation. Algorithm 2 depicts the whole procedure for applying the ensemble learning model to transform an unlabelled dataset into a labelled one.

Algorithm 2. Working of Ensemble Learning Perfect
1: Input: 2: D_{UL}: Unlabelled Dataset 3: FS: Feature set from Algorithm 1 4: Begin 5: Create an unfilled list D_{L} 6: Set W_1 = W_2 = 0:25 & W_3 = 0:50 7: for each data-entry d_{UL} in D_{UL} do 8: P_1 $= MBKmeans(FS(d_{UL}))$ 9: $P_2 = OP\ T\ ICS(FS(d_{UL}))$ 10: $P_3 = FCmeans(FS(d_{UL}))$ 11: $Cal.\ V\ using\ eq.\ 3$ 12: $if\ V > 0$: 5 $then$ 13: $Set\ V \widehat{=} 1$ 14: $else$ 15: $Set\ V \widehat{=} 0$ 16: $end\ if$ 17: $Append\ (d_{UL};$ V̂ $in\ D_L$ 18: $end\ for$ 19: $return\ DL$: $Labelled\ Dataset$

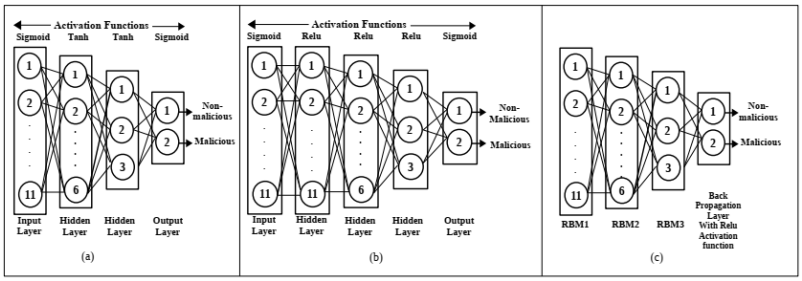

Figure 1. Neural network construction (a) LSTM NN (b) Multilayer Perceptron NN (c) DBN.

Labeled data may be produced with the help of the suggested ensemble model. In order to train various deep learning models, the created labelled dataset is put to use. Through performance analysis, we choose a model that is effective in detecting malicious assaults in an IIoT network using LSTM networks [24], MLPs [25], and DBNs [26]. Figure 1 depicts the underlying architectures from which the various deep neural network models were constructed. To detect unidentified network attacks at the edge layer, the trained DL model can be organised at the fog layer. This study uses a hybrid optimisation model to determine the LSTM's weight optimally, as will be shown below.

3.4.1 The HCPSO algorithm

In this paper, we suggest a fresh hybrid algorithm called the Hybrid Cat-Particle Swarm Optimization (HCPSO) algorithm. We integrate the CSO and PSO that are recognised as good metaheuristic algorithms. We employ the entire CSO scheme procedure in the HCPSO algorithm, with a few tweaks here and there. Similar to PSO, the algorithm stores both the global and local optimal positions. After that, we use of the specified dimension in searching mode, and the best new contender is picked to take its place. This hybridization attempts to achieve a faster-convergent algorithm without significantly increasing its execution time. All steps of the HCSPO algorithm are labelled as follows.

Then, in the range [0,1], generate a vector of N searchers' initial positions (X) and speeds (V).

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}, x_{kd} \in [0,1] \tag{11}$$

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1D} \\ v_{21} & v_{22} & \cdots & v_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \cdots & v_{ND} \end{bmatrix} \tag{12}$$

where, D is the number of items types.

1. Convert the position (X) into MBKP-MC solution term (Y) using Equation (14).

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1D} \\ y_{21} & y_{22} & \cdots & y_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{ND} \end{bmatrix} \tag{13}$$

$$y_{kd} = round(x_{kd} * (b_d - a_d)) \tag{14}$$

2. Verify all of the limitations. Make sure that all the solutions are an infeasible area which means all the solutions must meet the MBKP-MC constraints. Consider each solution's fitness value (total profit) and rank them accordingly. Divide the individuals into seeking and tracing modes.

3. Individuals who are actively seeking something. Create copies based on their own the best position C k Equation (15) and modify the selected dimension based on the best global solution C_g Equation (16).

$$x'_{j,d} = C_{k,d}, \quad d = 1,2, \ldots D \tag{15}$$

$$x'_{j,d} = C_{g,d} \pm SRD * r * C_{g,d} \tag{16}$$

4. If individuals are in tracing mode. Update the velocity and position based on PSO movement as formulated in Equation (17) - Equation (18)

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 \left(C_g(t) - X_i(t)\right) + c_1 r_1 (C_i(t) - X_i(t)) \tag{17}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{18}$$

Combine the cats in both the searching and tracing modes, making sure that no spots are beyond the range [0,1]. It is necessary to change the solution by means of Equation if it is larger than the search space (19).

$$x_k = \begin{cases} \dfrac{x_k - \min(x_k)}{\max(x_k) - \min(x_k)}, & if\ \min(x_k) < 0 \\ \dfrac{x_k}{\max(x_k)}, & if\ \max(x_k) > 1 \end{cases} \tag{19}$$

5. Convert the new position (X) into the MBKP-MC solution term (Y). Check all the constraints and then evaluate the fitness value.

6. Update the best individual position C_k and the best global position C_g.

7. Check the termination criterion. If the criterion is reached, then the algorithm is stopped and the final solution is C_g. But, if the criterion is not reached, go back to step 6.

4. Results and Discussion

4.1 The Evaluation Metrics

The following equations are used to evaluate the model's presentation using some of the most used metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$F - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{23}$$

Table 1 presents the validated results of proposed model. Figures 2 to 5 provide the graphical analysis of various metrics.

Table 1. Comparative investigation of test consequences

Algorithm	Precision	Recall	F-score	Accuracy
KNN	87.21	80.15	80.43	80.10
SVM	84.32	85.93	83.45	85.71
MLP	92.43	92.15	91.68	92.10
DBN	93.48	92.44	91.81	92.46
LSTM	90.21	89.54	89.03	89.52
Optimized Ensemble	**96.61**	**94.52**	**93.24**	**94.53**

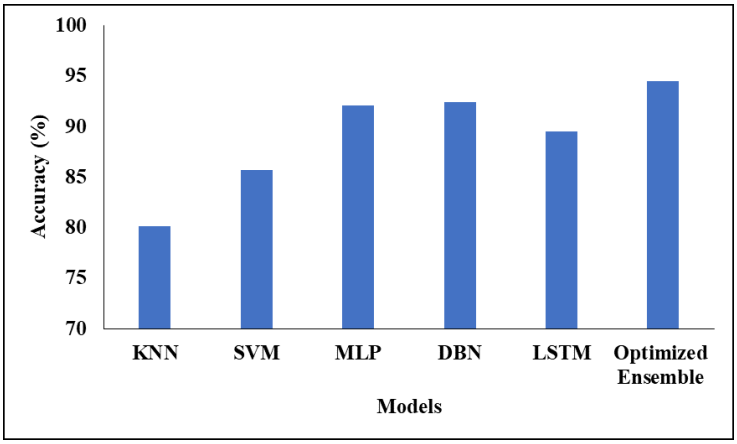

Figure 2. Graphical analysis in accuracy

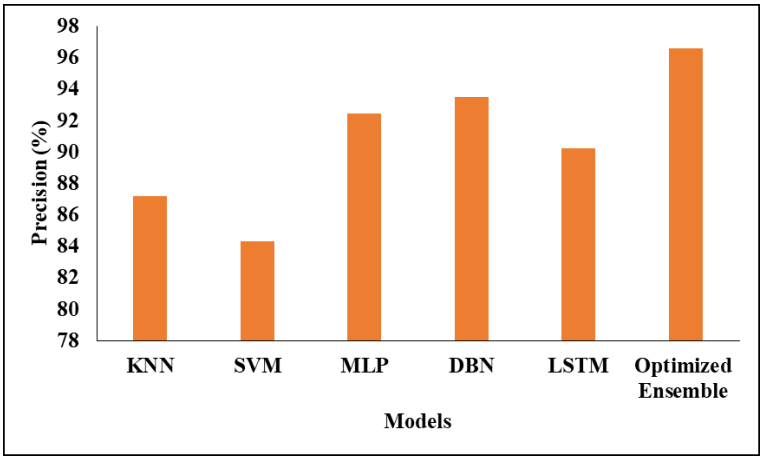

Figure 3. Precision analysis

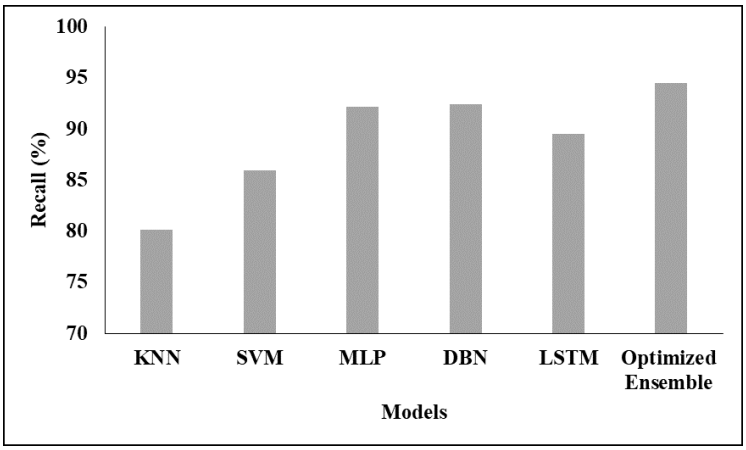

Figure 4. Comparative analysis of proposed model

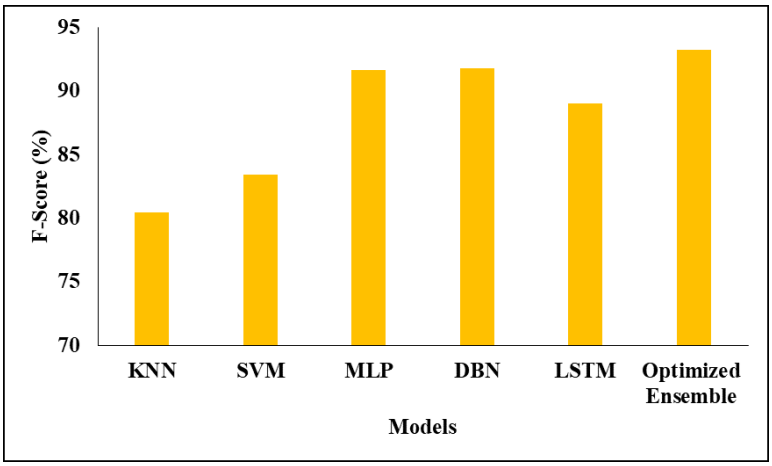

Figure 5. F1-score analysis

In the analysis of accuracy, KNN achieved 80%, SVM achieved 85%, MLP achieved 92%, DBN achieved 92%, LSTM achieved 89.52% and projected model achieved 94.53%. The reason for better presentation is that the weight of the LSTM is optimized by HCPSO. When comparing with all models, SVM achieved poor performance on precision, i.e., 84.32%, where the MLP, DBN, LSTM achieved nearly 90% to 94% of precision and finally, the proposed model achieved 96.61%. When the models are tested with recall and F-score, the KNN achieved 80%, SVM achieved 83% to 85%, MLP achieved 92%, DBN achieved 92%, LSTM achieved 89% and proposed ensemble model achieved 93% of F-score and 94.52% of recall.

5. Conclusion

We provide a model that may transform an unlabeled network dataset into a labelled one, allowing for the prediction of previously undiscovered attacks. The AI-based ensemble model that aided the intelligence layer in predicting the output label is evaluated using accuracy, F1-score. The projected ensemble learning model converts the dataset into a labelled dataset so that it may be used to train a DL model. Improved versions of the LSTM, MLP, and DBN deep learning models were used to increase classification accuracy in the study. With an attack detection accuracy of 95% on the analysed dataset, the results show that optimised LSTM performs better than the other two DL models in identifying malicious assaults in an IIoT network. To identify new threats, the proposed unsupervised ensemble-based learning algorithm analyses unlabeled IIoT network data. In a fog computing setup, this concept may be used in the cloud. In order to use deep learning models for network intrusion detection, the proposed method labels network traffic. The trained network may be installed at the fog layer to analyse the network traffic of edge devices and identify them, with frequent updates in the cloud to account for new assaults. The stress on fog and on power- devices may be decreased by employing a fog computing architecture. Implementing the suggested model on a real-world IoT network using a fog computing architecture would allow us to further investigate its efficacy and complexity. The study also introduces a hybrid optimisation algorithm for pinpointing the optimal LSTM weight when it comes to intrusion detection. When trained on the labelled dataset provided by the proposed method, the improved LSTM outperforms the other models with a finding accuracy of 95%, as exposed in the research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Alansari, N. B. Anuar, A. Kamsin, and M. R. Belgaum, “A systematic review of routing attacks detection in wireless sensor networks,” *PeerJ Computer Science*, vol. 8, Article ID: e1135, 2022.

https://doi.org/10.7717/peerj-cs.1135.
[2] R. Vatambeti, and V. K. Damera, "Gait based person identification using deep learning model of generative adversarial network," *Acadlore Trans. Mach. Learn.*, vol. 1, no. 2, pp. 90-100, 2022. https://doi.org/10.56578/ataiml010203.
[3] M. Ezhilarasi, L. Gnanaprasanambikai, A. Kousalya, and M. Shanmugapriya, "A novel implementation of routing attack detection scheme by using fuzzy and feed-forward neural networks," *Soft Comput.*, vol. 27, pp. 4157-4168, 2023. https://doi.org/10.1007/s00500-022-06915-1.
[4] C. Jothi Kumar, V. Deeban Chakravarthy, K. Ramana, et al. "OTP-ER: An ordered transmission paradigm for effective routing in IoT based wireless sensor networks," *Opt. Quant. Electron.*, vol. 54, Article ID: 456, 2022. https://doi.org/10.1007/s11082-022-03837-y.
[5] R. Gupta, N. K. Jadav, H. Mankodiya, M. D. Alshehri, S. Tanwar, and Sharma, R. "Blockchain and onion-routing-based secure message exchange system for edge-enabled IIoT," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp.1965-1976, 2023. https://doi.org/10.1109/TII.2022.3191444.
[6] R. Vatambeti, N. S. Divya, H. R. Jalla, and M. V. Gopalachari, "Attack detection using a lightweight blockchain based elliptic curve digital signature algorithm in cyber systems," *International Journal of Safety and Security Engineering*, vol. 12, no. 6, pp. 745-753, 2022. https://doi.org/10.18280/ijsse.120611.
[7] J. He, K. Y. Lin, and Y. Dai, "A data-driven innovation model of big data digital learning and its empirical study," *Inf. Dyn. Appl*., vol. 1, no. 1, pp. 35-43, 2022. https://doi.org/10.56578/ida010105.
[8] E. Garcia Ribera, B. Martinez Alvarez, C. Samuel, P. P. Ioulianou, and V. G. Vassilakis, "An intrusion detection system for RPL-based IoT networks electronics," *Electron.*, vol. 11, no. 23, Article ID: 4041, 2022. https://doi.org/10.3390/electronics11234041.
[9] N. S. Divya, V. Bobba, and R. Vatambeti, "An adaptive cluster based vehicular routing protocol for secure communication," *Wireless Pers. Commun.*, vol. 127, pp. 1717-1736, 2022. https://doi.org/10.1007/s11277-021-08717-4.
[10] E. Gyamfi, and A. Jurcut, "M-TADS: A multi-trust DoS attack detection system for MEC-enabled industrial loT," In 2022 IEEE 27th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, Paris, France, pp. 166-172. 2022. IEEE. https://doi.org/10.1109/CAMAD55695.2022.9966900.
[11] N. Bhandary, R. S. Shalakha, P. Honnavalli, and E. Sivaraman, "An enhanced malware detection approach using machine learning and feature selection", In 2022 3rd International Conference on Electronics and Sustainable Communication Systems, Coimbatore, India, pp. 909-914 2022. https://doi.org/10.1109/ICESC54411.2022.9885509.
[12] B. B. Behera, R. K. Mohanty, and B. K. Pattanayak, "A deep fusion model for automated industrial iot cyber attack detection and mitigation," *International Journal of Electrical and Electronics Research*, vol. 10, no. 3, pp. 604-613. https://ijeer.forexjournal.co.in/papers-pdf/ijeer-100332.pdf.
[13] C. Chethana, P. K. Pareek, V. H. C. de Albuquerque, A. Khanna, and D. Gupta, "Improved domain generation algorithm to detect cyber-attack with deep learning techniques," In 2022 IEEE 2nd Mysore Sub Section International Conference, Mysuru, India, pp. 1-8, 2022. https://doi.org/10.1109/MysuruCon55714.2022.9972526.
[14] B. B. Behera, R. K. Mohanty, and B. K. Pattanayak, "Attack detection and mitigation in industrial IoT: An optimized ensemble approach," *Specialusis Ugdymas*, vol. 1, no. 43, pp. 879-905, 2022.
[15] C. Pedroso and A. Santos, "Dissemination control in dynamic data clustering for dense IIoT against false data injection attack," *International Journal of Network Management*, vol. 32, no. 5, 2022. https://doi.org/10.1002/nem.2201.
[16] J. Ryu and S. Kim, "Reputation-based opportunistic routing protocol using Q-Learning for MANET attacked by malicious nodes," *IEEE Access*, 2023. https://doi.org/10.1109/ACCESS.2023.3242608.
[17] I. Obeidat, A. Mughaid, and O. Alofishat, "A new detection jamming attack model on VANET ad hoc networks," *Research Square*, 2023. https://doi.org/10.21203/rs.3.rs-2371703/v1.
[18] S. Rabhi, T. Abbes, and F. Zarai, "IoT routing attacks detection using machine learning algorithms," *Wireless Pers. Commun.*, vol. 128, no. 3, pp. 1839-1857, 2023. https://doi.org/10.1007/s11277-022-10022-7.
[19] M. Malik and M. Dutta, "Feature engineering and machine learning framework for DDoS attack detection in the standardized Internet of things," *IEEE Internet of Things Journal*, 2023. https://doi.org/10.1109/JIOT.2023.3245153.
[20] R. Alghamdi and M. Bellaiche, "A cascaded federated deep learning based framework for detecting wormhole attacks in IoT networks," *Computers & Security*, vol. 125, Article ID: 103014, 2023. https://doi.org/10.1016/j.cose.2022.103014.

[21] F. K. Örs and A. Levi, "Data driven intrusion detection for 6LoWPAN based IoT systems," *Ad Hoc Networks*, vol. 143, Article ID: 103120, 2023. https://doi.org/10.1016/j.adhoc.2023.103120.

[22] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-IIoTID: A connectivity-and device-agnostic intrusion dataset for industrial internet of things," *IEEE Internet of Things Journal*. vol. 9, no. 5, pp. 3962-3977, 2022. https://doi.org/10.1109/JIOT.2021.3102056.

[23] G. Rani, M. G. Oza, V. S. Dhaka et al. "Applying deep learning-based multi-modal for detection of coronavirus," *Multimedia Systems*, vol. 28, pp. 1251-1262, 2022. https://doi.org/10.1007/s00530-021-00824-3.

[24] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," In 2016 international conference on platform technology and service (PlatCon), Jeju, Korea (South), *IEEE*, pp. 1-5, 2016. https://doi.org/10.1109/PlatCon.2016.7456805.

[25] J. Franklin, "The elements of statistical learning: Data mining, inference, and prediction," *The Mathematical Intelligencer*, vol. 27, pp. 83-85, 2005. https://doi.org/10.1007/BF02985802.

[26] G. E. Hinton, S. Osindero, Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527-1554, 2006. https://doi.org/10.1162/neco.2006.18.7.1527.