Mesopotamian journal of Computer Science
Vol. 2025, **pp**. 105–114
DOI: https://doi.org/10.58496/MJCSC/2025/007 ; ISSN: 2958-6631
https://mesopotamian.press/journals/index.php/cs

Research Article

A Deep Learning Approach for Identifying Malicious Activities in the Industrial Internet of Things

Mohammed Amin Almaiah[1,*], Fuad Ali El-Qirem[2], Rami Shehab[3], Khaled Sulieman Momani[4]

[1] *King Abdullah the II IT School, The University of Jordan, Amman 11942, Jordan .*

[2] *Faculty of Architecture and Design, Al-Zaytoonah University of Jordan, Amman 11733, Jordan.*

[3] *Vice-Presidency for Postgraduate Studies and Scientific Research, King Faisal University, Al-Ahsa, Saudi Arabia..*

[4] *Department of Educational Administration, Faculty of Educational Sciences, Ajloun National University, Ajloun, Jordan.*

ARTICLE INFO

Article History

Received 26 Feb 2025
Revised 25 Mar 2025
Accepted 23 Apr 2025
Published 22 May 2025

Keywords

Industrial Internet of Things (IIoT)

Deep Learning

Intrusion Detection Systems (IDS)

Cybersecurity

Anomaly Detection

ABSTRACT

Data-driven decision-making, real-time connectivity, and automation have transformed industrial operations with the Industrial Internet of Things. However, the integration also introduces substantial cybersecurity vulnerabilities, making IIoT networks a prime target for malicious activities. Cyber threats are evolving and becoming more sophisticated, which makes traditional security mechanisms inadequate. An approach using deep learning to detect malicious activities in IIoT environments is examined. It is investigated whether Deep Feed Forward neural networks, autoencoders, and convolutional neural networks are effective at detecting anomalies and mitigating cyber threats. NSL-KDD and UNSW-NB15 benchmark datasets are used to evaluate the proposed model's accuracy, precision, and detection rates. In addition to strengthening IIoT security, deep learning techniques can also ensure the resilience of industrial infrastructure.

1. INTRODUCTION

With the Industrial Internet of Things (IIoT), modern industries are becoming more connected, automated, and efficient [1]. As it interconnected ecosystem becomes increasingly vulnerable to malicious activities and cyberattacks, it also poses significant cybersecurity challenges. The dynamic and complex threat landscape of the IIoT environment often makes traditional security measures inadequate. To identify and mitigate malicious activities, deep learning has emerged as a powerful tool. Using advanced neural networks, deep learning approaches offer innovative solutions for detecting anomalies, predicting threats, and strengthening IIoT security frameworks. A deep learning approach is used to enhance cybersecurity resilience and ensure the sustainability of IIoT networks. As a result of IoT implementation, a new generation of communication and information technology has emerged. These technologies have been used in a number of vital industries to provide cost-effective, automated, sustainable, and smart solutions [2], [3]. Industry Internet of Things (IIoT) is the result of the extensive integration of IoT into the industrial and manufacturing domains. Several industrial and manufacturing sectors have implemented the Internet of Things, leading to the Industrial Internet of Things (IIoT). Medical devices, robotics, and software-defined manufacturing are also part of the IIoT in addition to industrial applications [4], [5]. The IIoT provides better quality of service and QoE for consumers, as a result of the IIoT. With sensors, smart actuators, and remote control, it's easier to monitor, manage, and control physical infrastructure in agriculture, healthcare, manufacturing, and transportation [6]. Industry 4.0 enables industrial devices to make real-time decisions using big data and analytics by transforming cyber-physical systems and production processes [7].

With NIDS, industrial IoT networks can be protected against cyberattacks and made secure and private [8]. A NIDS analyses network traffic for suspicious behaviour, and if it detects any, an alarm is raised. By comparing cyberattack signatures with incoming traffic patterns, typical intrusion detection systems can detect cyberattacks. Detection of traffic patterns depends on whether they match an attack signature. It kind of IDS is very effective at detecting known cyberattacks. Artificial

*Corresponding author. Email: m.almaiah@ju.edu.jo

intelligence (AI) has attracted an enormous amount of interest from academia and industry because of its ability to perform tasks more efficiently and intelligently than traditional methods, as well as the fact that it can detect cyberattacks previously undetected by IDSs [9]. As part of the cybersecurity field, deep learning techniques have been applied to identify and mitigate a number of cyberattacks [10], providing impressive detection results [11]. With IIoT devices potentially tracking user behaviour through hardware and software, it is crucial to design policies and technical solutions that protect privacy, safety, and freedom. Every day, millions of devices connect to the Internet, opening up the possibility of multifaceted cyberattacks. It is common for IIoT devices to be hacked, accessed without authorization, and have their data stolen. Cybersecurity risks are associated with devices and networks connected to the Internet of Things [12]. Mitigating these challenges is a research priority. Devices and networks connected to the internet of things are often targeted by attacks, including distributed denial-of-service (DDoS) attacks, keylogging attacks, and data theft. Most attacks on IIoT devices and networks are believed to be caused by botnets[13]. Each device in a botnet runs one or more bots and is connected to the Internet. A botnet's computational power increases as more IIoT devices are infected, making it more powerful to carry out larger attacks [14].

2. RELATED WORK

IDS-based deep learning algorithms are proposed by the author [14]. Using deep belief networks, an IDS has been developed. We used disjoint datasets to train and validate the proposed method. In the second model, the chosen DL algorithm is trained on unlabeled data sets, and network traffic is analyzed over time. Additionally, an IDS based on deep learning was proposed for the IIoT [15]. With a deep feed-forward neural network and a deep autoencoder, it can learn malicious traffic characteristics. Modelling was done based on TCP/IP packet data. Datasets from NSL-KDD and UNSW-NB15 were useful for detecting cyberattacks on industrial IoT systems[16]. In several papers [17], sparse and noisy deep autoencoders are proposed and then differentiated by using deep learning networks. Our proposed method evaluates whether it is able to identify attacks against IIoT systems based on remote telemetry collected from gas line systems. An industrial IoT network needs to be protected from deep random neural networks, according to the author of [18]. A dataset from UNSW-NB15 was used as validation for the proposed method. In accordance with the system results, detection accuracy was 99.54 percent, and false alarms were low. According to the author, DL-based IDS approaches can be applied to industrial IoT systems [19], [20]. An intrusion detection model using multi-CNN fusion was built by combining several convolutional neural networks. The NSL-KDD dataset was evaluated for its ability to detect attacks on industrial IoT systems. Author [21] the absence of datasets required to build security solutions in smart environments. For the purpose of simulating the traffic of IoT and IIoT networks under various types of cyberattacks, a testbed framework was developed, and an experimental environment was created. An experimental testbed containing all types of attacks, including normal attacks, generates data for TON-IoT. The authors employed a variety of machine learning and deep learning models to analyze the produced dataset in order to encourage future research. DL has been proposed as a method for identifying IIoT cyberattacks [22]. Particle swarm optimization was used to optimize neural network hyperparameters. An attack detection model was developed using deep random neural networks during the training phase. This model was found to be effective based on the results. The author [23] uses SDN for intrusion detection and illustrates a 95% testing rate for the IoT. An SDN-based approach to prevent the detection of hosts after infection is described in [24]. Several datasets have been considered for implementation, including ISOT and CTU-13. A 99.2% detection accuracy is achieved when MLP is taken into account. A 99.2% detection accuracy is achieved when MLP is taken into account [25]. A proposed scheme for identifying botnet attacks in [26] used network flow capability. Detection of botnets was based on Naive Bayes (NB), J48, and Bayesian models [31]. Dartmouth's wireless network is used to collect data using detectors. The author describes a system that memorizes harmful network behaviours and detects and prevents botnet infections [26].
In the KDD99 dataset, 98% of the detections were accurate using the devised approach. The Author [27], DL-based algorithms (e.g. LSTM) and IoT-based algorithms were proposed for detecting botnet attacks using IoT [28]. Among the IoT devices analyzed from various manufacturers using the N-IoT 2018 datasets, a detection rate of 99.90% was achieved

3. METHODOLOGY

The study examined feed-forward neural networks, gated recurrent units, recessive neural networks, and convolutional neural networks.

3.1. Deep Feed-Forward Neural Network

Deep Feed-forward Neural Networks (DFFNN) are used in many currently available deep learning architectures. This type of neural network transmits only forward-looking information. Machine learning applications rely heavily on it [29]. A hidden layer is created based on the output of each hidden layer node, with weights and biases based on the input from the hidden layer. A backpropagation method and loss function are used to adjust the weights to obtain optimal results. The backpropagation and loss functions of a network are used to adjust weights to maximize performance. There are three entries of historical data on the input layer, three entries on the hidden layer, and one entry on the output layer. A DFFNN,

unlike an RNN, only feeds forward at the output and does not give feedback to the network. The limitations of DFFNN make it unsuitable for time series forecasting applications such as detecting and classifying faults in secondary distribution networks.

An essential noteworthy learning model is the Feed-Forward neural memory or the multilayer Perceptrons (MLPs). There is too much unpleasantness in the inspiration of a Feed-Forward association. A classifier can be used to characterize data using $y = f^{x}(x)$, for instance. In the Feed-Forward network, the $y = f(x)$ gauge is used to determine the potential gains of the limits. By Feed-Forward models, information travels between x and y. Therefore, momentary assessments are key in describing f and yielding y. In disjointed neural connections, processing relationships are connected by lightening up neural connections. A DFFNN is an ANN made up of a perceptron, a few mystery centres, and a yield neuron with no cycles [30]. As neural organization variables may act arbitrarily, this calculation-based information approach places formats in minimum arrangements with no help from standards [31]. When solo techniques are pre-prepared, and an AE is explicitly used to construct initiation details, union rates and resulting results can be improved [31].

3.2. Deep Auto-Encoder (DAE)

Using unaided DAEs, a neural organization is performed in a feed-forward manner [32]. Information $(x \to \hat{x}), x$ is meaningful when it is compared to an assortment of information $(\widehat{x})$

The information hubs depict some disguised units of non-straight starting ascribed as vectors. x^{i}. The separate components of the brain use fewer neurons than the hubs of information so that they can acquire smaller amounts of information. As a result, the main credits are known, the spatial dimensions are reduced, and perspectives of the information become apparent. Eq. (1) shows the yield layer. $\hat{x}^{i}$ as a local representation of the information layer at the end of the process

$$T(t) = \frac{1 - e^{-2t}}{1 + e^{-2t}} \tag{1}$$

AE calculation relies on encoders and decoders [36], and using deterministic planning and an encoder technique FO [37], the information vector x^{i} is transformed into a secret layer representation z_i, thereby reducing the number of codes and presenting a dimension x_i As shown in Equation 2.

$$f\theta(x^{i}) = T\left(W_{x^{i}} + b\right) \tag{2}$$

In this graph, $W = d^{0}xd^{h}, d^{h}$ represents the weighted network, T represents the Tanhinitiation utility, θ, $[W, b]$ represents the plan boundaries, and $(d^{0} < d^{h})$ represents the disguised level neurons. In the figure, the hidden layer's output is projected. A deterministic plot determines $(g^{\theta'})$ interpretation method, which, in Equation 3, is used to build $(g^{\theta'})$ contribution as an estimate x^{i}.

$$g\theta'(x^{i}) = T\left(W'_{z^{i}} + b'\right) \tag{3}$$

$d^{0}xd^{h}$, represents the weight framework $[W', b']$ by θ addresses planning boundaries in W', and b represents predispositions. Once the packed representation has been transformed to fit the mysterious surface, it is used for the reconstruction of the first data. Norm or little cluster preparation sets (S) undergo a rearranging called a change botch when their preparation interactions are described in Eqs. 4 and 5.

$$E(x, \hat{x}) = \frac{1}{2}\sum_{i}^{s} \left|\left|x^{i} - \hat{x}^{i}\right|\right|^{2} \tag{5}$$

$$\theta = [W, b] = argmin_{\theta} E[x, \hat{x}) \tag{6}$$

3.3. Feature Selection

Based on the feature $f_k \in F', k - 1,2, \ldots, p$, we can extract the feature subset $F' \in F$. Considering that $\psi_1, \psi_2, \ldots . \psi_n$ is a potential value for $\psi_1, \psi_2, \ldots . \psi_n$, the important measure of F' can be expressed as Equation 7.

$$Imp(f) = -\sum_{i=1}^{n} \frac{\psi_i}{Z} . \log\left(\frac{\psi_i}{Z}\right) - \sum_{i=1}^{n} (1 - \frac{\psi_i}{Z}) . \log\,(1 - \frac{\psi_i}{Z}) \tag{7}$$

$\psi(x) = \frac{1}{n}\sum_{i=1}^{n}\{\prod'_F m_i \,.\, k\left(\frac{x - X_i}{\sigma}\right)\}, z$ is the sum of potential values

Feature vectors are measured by entropy based on how much information they contain. A feature with a high level of information has a lower entropy. Position potentials will tend to be similar if an object is uniformly distributed, and feature importance will tend to be similar if it is uniformly distributed. Asymmetric distributions, on the other hand, have a lower entropy potential. An optimal entropy for a data field is defined by Equation 8 when calculating its potential entropy.

$$Imp(f)_{opt} = imp(f)_{\sigma = \sigma_F,} \tag{8}$$

This paper employs the nonparametric kernel density estimation method proposed by Hall P for calculating σ. There are σ standard deviations between the upper and lower density of each direction. The calculation requires another parameter, m_i. Potential matrices can be calculated from the potential functions in generalized data fields. A feature's importance in supervised learning problems depends on the S_w and S_b values of the class potential.

$$m_i = \frac{S_{wi}^{\psi}}{S_{bi}^{\psi}} \tag{9}$$

Anywhere

$$S_{wi}^{\psi} = \frac{1}{n}\sum_{j=1}^{n}(\psi(X_i) - \psi_j(x)\left(\psi(X_i) - \psi_j(x)\right)^T \tag{10}$$

According to Equation 11, the spatial distribution matrix between classes S_{bi}^{ψ}, B is as follows:

$$S_{bi}^{\psi} = \frac{n_i}{n}\big(\psi(X_i) - \psi(X)\big)\big(\psi(X_i) - \psi(X)\big)^T \tag{11}$$

As shown in Equation 12, there are two distribution matrices for features.

$$S_t^{\psi} = S_w^{\psi} + S_b^{\psi} = \frac{1}{n}\sum_{i=1}^{n}(\psi(X_i) - \psi_j(x)\left(\psi(X_i) - \psi_j(x)\right)^T \tag{12}$$

Where $\psi(X) = \frac{1}{n}\sum_{i=1}^{n}(\psi(X_i), \psi_j(x) = \frac{1}{n}\sum_{i=1}^{n}(\psi(X_i), j = 1, \dots, c$. A non-negative matrix would be S_w^{ψ} and S_b^{ψ} based on the definition above. Normalization refers to normalizing a given feature based on deviations from it. As a result of linear transformation, the raw data is transformed into [0-1] values. As shown in Equation 13, the translation function is equal to

$$f = \frac{f - min}{max - min} \tag{13}$$

Data samples are classified as max or min according to their maximum and minimum values, respectively. A mass vector $M = \{m_1, m_2, \dots . m_i, m_n\}$ is created by multiplying m_i by the number of features, and MI by the weight of each feature. In this case, $\sigma = \{\sigma_1, \sigma_2 \dots \sigma_i, \sigma_n\}, \sigma_i$ is a vector representing the factors affecting the feature. Based on Equation 14, the mean of every sample in all directions represents the sample's potential value.

$$\psi(X) = \frac{1}{n}\sum_{i=1}^{n}(\psi(X_i) \tag{14}$$

$\psi(X_i) = \prod_F' m_j . k\left(\frac{x_j - x_{i,j}}{\sigma_j}\right), j$ represents the i^{th} sample from j^{th}. Utilizing the hierarchical clustering method, the best subset of features is obtained after determining the importance of the features. It is possible to describe the distance between the selected feature subset. F_0 and label class $C, S_b(C;\ F')$ by using Equation 15, which sums the distance between the selected features and the label class.

$$S_b(C; F' = \sum_{i=1}^{n} S(C_i; f) \tag{15}$$

Because alternative features correlate with selected features:

$$S(f) = \sum_{f \epsilon F'} CU(f, s) \tag{16}$$

Within the class distance, Equation 17 defines the update function.

$$S_w(F', s) = S_w(F') + s(f) \tag{17}$$

Furthermore, F' must be considered in terms of its size. Classifiers with smaller feature subsets are generally more robust because they have fewer selected features. According to the above analysis, each candidate feature has a set of evaluation functions defined by Equation 18.

$$J(F') = \frac{S_b(C; F', s)}{|F'| + S_w(F', s)} \tag{18}$$

In feature subsets, $|F'|$ represents the feature number. $J(F')$ is more useful for classification if its value is higher, meaning there is a closer correlation between the new feature and the class label. Correlations between selected feature subsets and new candidate features are also considered during evaluation. There will be too much correlation between candidate features and existing feature subsets, which means that it is redundant and unnecessary to include this feature.

There is evidence that new features enhance rather than detract from a feature subset. This results in improved classification accuracy, shortened feature subsets, and guaranteed feature subsets.

3.4. Datasets Used

Model accuracy is evaluated by using data sets. Data quality is an essential component of Network Intrusion Detection Systems (NIDS).

We will examine three sets of data in this analysis: KDD Cup'99, NSL-KDD, and UNSW-NB15. Here are more details about these features.

3.4.1. KDD Cup'99 Data set

As a result of the 1998 dataset, DARPA created KDD'99 in 1999. Preprocessing is performed on 41 features per network connection. KDD'99 comprises four categories of features: A basic feature set (1-9), a content feature set (10-22), a time-based traffic feature set (23-31), and a host-based traffic feature set (32-41). In comparison to other data sets, KDD'99 [33]   contains 4,898,430 records. There are four general types of attacks: DoS, Remote-to-Local, Unauthorized Root Access, and Probe.d. Various data mining techniques were used in KDD'99 in order to detect network intrusions. During KDD Cup'99, intrusion detection systems (IDSs) mainly used KDD Cup'99 data. KDD's performance is significantly affected by two crucial issues with its data set, according to the arithmetical analysis. The KDD data set has numerous duplicate records, which is a major problem. There is approximately a 78% and 75% duplicate data set in the train and test, respectively. In lieu of numerous records, duplicated records may produce partial learning algorithms. The algorithm will, therefore, no longer learn infrequent records. This will prevent the algorithm from learning rare records.

3.4.2. NSL-KDD Dataset

NSL-KDD was proposed as a data set containing a selection of records from the entire 1999 KDD Cup data set. Compared to KDD Cup'99, NSL-KDD has the following advantages: Classifiers won't be biased against repetitive records as a result of excluding irrelevant records from the train set. KDD data indicates that the percentage of records with difficulty records depends on how many records are selected from the difficulty record. As a result, different ML (Machine Learning) techniques produce different percentages of classification correct. In this way, it becomes possible to conduct structured, comprehensive evaluations of ML approaches [34].

- Rather than randomly selecting small segments from a large dataset, experiments can be performed on logically numbered sets. Therefore, different evaluations will produce similar results.

3.4.3. UNSW-NB15 Dataset

A new dataset, UNSW-NB15, was published in 2015[35]. The KDD'99 dataset contained 14 attack types, whereas the current dataset contains nine attack types. There are 49 features and an assortment of normal and attacked activities in this collection, as well as 25,40,044 records with class labels. The number of normal records found is 2,214,876, and the number of attacked records is 3,214,283. Among UNSW-NB15's features, there are six main categories: Basics, Flows, Time, Contents, Additional Generated, and Labeled Features. Feature 36-40 is referred to as a general-purpose feature. Connectivity is defined as 41-47 features. There are also nine types of attacks included in the UNSW-NB15 dataset: analysis, fuzzers, backdoors, denial-of-service attacks, reconnaissance, generics, shellcodes, and worms.

3.5. Evaluation Metrics

A number of model parameters were evaluated in the evaluation process, including accuracy, precision, sensitivity (recall), F1, specificity, and precision-recall. Using Decision Tree models and Machine Learning models, cybersecurity intrusions were classified and detected multi-dimensionally. Autoencoders were used in Industry 4.0 WSNs for cyber intrusion detection. RF and LR benchmark models were compared with those implemented in the proposed framework to assess how well they performed. An evaluation of binary classification models is based on their specificity and precision-recall curves. RF models, Decision Trees, and MLPs were used to classify multi-dimensionally using accuracy, precision, and sensitivity. A number of metrics were considered when classifying binary data using Autoencoder and LR models, such as specificity, precision-recall curve metrics, accuracy, precision, sensitivity, and F1 score:

True Positives (TP): Upon completion, how many tuples seem intrusive?

True Negatives (TN): After detection, the number of valid tuples found.

False Positives (FP): When the detection process is complete, the number of safe tuples is recognised as intrusions.

False Negatives (FN): As a result of the detection process, how many dangerous tuples are detected?

Classification models are often evaluated by their accuracy. A model's accuracy is measured by how many cases are correctly predicted. As a mathematical expression, it is represented by A and can be calculated by equation 19.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

A classification model's precision can be measured by how well it performs. The model's accuracy is determined by counting the true positives out of all positive predictions or by counting the true positives plus fake positives. Equation 20 can be calculated by using P as the mathematical representation.

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

Its sensitivity measures an effective classification model. Alternatively, you can refer to this number as the true positive rate or recall rate. This parameter measures how well the model can identify every positive case in the dataset. Equation 21 can be used to calculate it mathematically—R represents it.

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

A classification model's F1 score indicates how well it does when choosing between two options. F1 scores are helpful when accuracy and recall are different. By using equation 22, you can calculate the F1-Score mathematically.

$$F1 - Score = 2 \times Precision \times \frac{Recal}{Precision + Recall} \tag{22}$$

Binary classification is evaluated based on specificity, a performance metric. This metric measures how accurate a model is at identifying negative instances out of all possible negatives. Equation 23 shows how to calculate it mathematically by using S.

$$Specificity = \frac{TN}{TN + FP} \tag{23}$$

4. RESULTS AND DISCUSSION

Figures 1 and 2 provide a comparison of the proposed model's performance across various metrics. These figures are based on NSL-KDD data as well as UNSW-NB15 data. The proposed model is effective at predicting and categorizing attacks within IIoT networks, proving its significance and applicability.

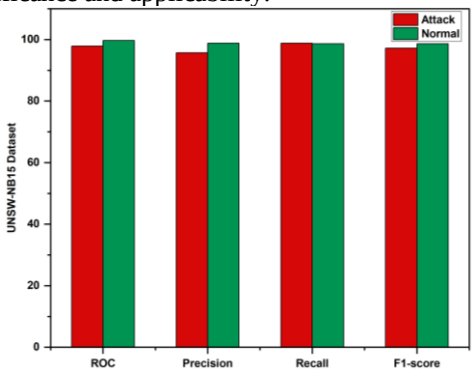

Fig. 1. Proposed method evaluation (for NSL-KDD).

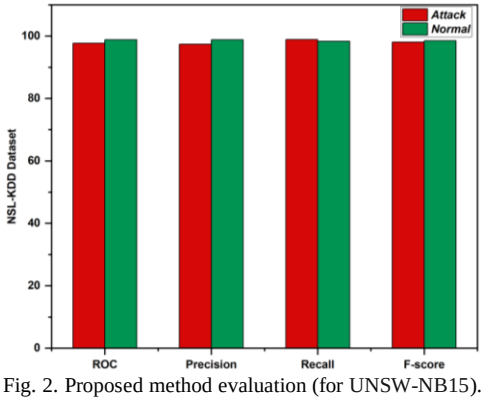

Fig. 2. Proposed method evaluation (for UNSW-NB15).

According to Figure 3, the proposed model performed well when applied to the datasets; the detection rate was high, and the false positive rate was low. Comparing the model with UNSW-NB15, the NSL-KDD dataset yielded 97.5% precision, 98% detection rate, and 2.5% and 2% FPR, respectively.

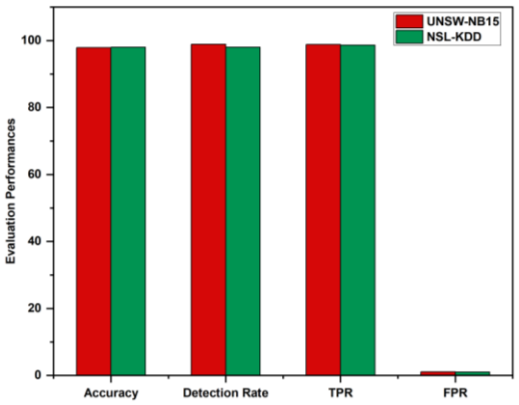

Fig. 3. Evaluation of performances for two datasets.

The proposed approach is compared with a number of established approaches in Figure 4 to illustrate how feature selection impacts the effectiveness of the classification algorithm. A reduced UNSW-NB15 dataset shows that the proposed model and other models produce similar results. Compared to other approaches, the proposed method has a smaller number of false positives (FPR). Compared to the modified KNN model, which was the next-best method, the network intrusion detection system has an overall accuracy of 97.9%, 0.2% better than the modified KNN model.

A lower FPR of 2.5% is also displayed by the proposed method, outperforming other classifiers. Based on all evaluation metrics using UNSW-NB15, the proposed approach consistently outperforms. Several factors contribute to its slightly improved accuracy, including a robust feature selection mechanism and a rule-based fitness evaluation.

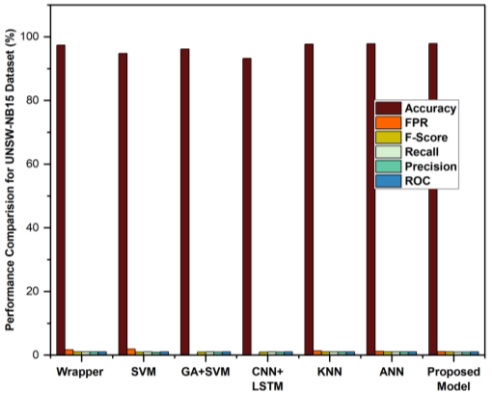

Fig. 4. A comparison of UNSW-NB15's performance.

A comparison of the recognition rate and false positive rate using NSL-KDD's dataset indicated that our proposed system outperformed other models. The developed scheme resulted in DR and FPR of 99 percent and 1.8 percent, respectively. With the first four models, destructive events can be identified rationally using a feature selection process. The F-SVM solves linear and nonlinear properties of data using shared information and is paired with SVMs to detect attacks. Nonetheless, IDS needs to refine its search strategy to make it more effective. Using PCA, CVT, and TANN reduced the data measurements. There is a 98 percent efficiency rate for our proposed model, which is higher than other existing models.

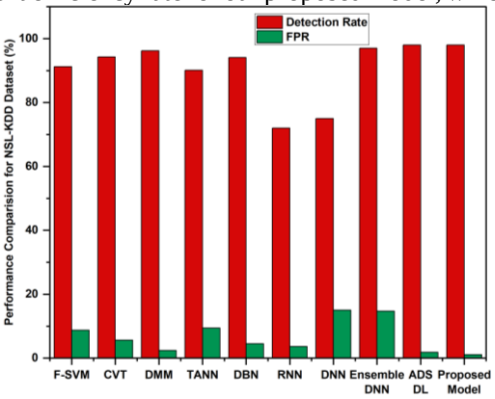

Fig. 5. A comparison was made between the proposed model and eight other classifiers.

5. CONCLUSION

The study illustrates the effectiveness of deep learning-based techniques in protecting IIoT networks against evolving cyber threats. Using advanced neural networks, our proposed model detects intrusions more effectively than traditional security methods. Based on experimental results on data sets such as NSL-KDD and UNSW-NB15, the model proves to be a reliable security solution for IoT. Additionally, the study emphasizes the importance of feature selection for classification efficiency. IoT devices with limited resources should be eligible for resource-constrained federated learning, lightweight deep learning approaches for privacy-preserving intrusion detection, and real-time threat adaptation. To ensure the sustainability and resilience of industrial automation systems in the face of ever-evolving cyber threats, IIoT cybersecurity must be strengthened.

Funding

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Grant No. 55554).

Conflicts Of Interest

The author's affiliations, financial relationships, or personal interests do not present any conflicts in the research.

Acknowledgment

The authors extend appreciation to the institution for their unwavering and Special thanks to Al-Zahraa University for Women, University of Kerbala, ,AlMustaqbal University, Al-Ameed University and Alsafwa University College Dean's University support and encouragement during the course of this research.

References

[1] P. Rani and R. Sharma, “Intelligent transportation system for internet of vehicles based vehicular networks for smart cities,” *Computers and Electrical Engineering*, vol. 105, p. 108543, 2023.

[2] S. Bacha, A. Aljuhani, K. B. Abdellafou, O. Taouali, N. Liouane, and M. Alazab, “Anomaly-based intrusion detection system in IoT using kernel extreme learning machine,” *J Ambient Intell Human Comput*, vol. 15, no. 1, pp. 231–242, Jan. 2024, doi: 10.1007/s12652-022-03887-w.

[3] R. Alanazi and A. Aljuhani, “Anomaly Detection for Industrial Internet of Things Cyberattacks,” *Computer Systems Science and Engineering*, vol. 44, no. 3, pp. 2361–2378, 2023, doi: 10.32604/csse.2023.026712.

[4] S. Jain and K. Chandrasekaran, “Industrial Automation Using Internet of Things:,” in *Advances in Information Security, Privacy, and Ethics*, P. Ahlawat and M. Dave, Eds., IGI Global, 2020, pp. 28–64. doi: 10.4018/978-1-7998-0373-7.ch002.

[5] P. Rani and R. Sharma, “Intelligent Transportation System Performance Analysis of Indoor and Outdoor Internet of Vehicle (IoV) Applications Towards 5G,” *Tsinghua Sci. Technol.*, vol. 29, no. 6, pp. 1785–1795, Dec. 2024, doi: 10.26599/TST.2023.9010119.

[6] J. Simon, N. Kapileswar, P. K. Polasi, and M. A. Elaveini, “Hybrid intrusion detection system for wireless IoT networks using deep learning algorithm,” *Computers and Electrical Engineering*, vol. 102, p. 108190, 2022.

[7] Q. Abu Al-Haija and S. Zein-Sabatto, “An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks,” *Electronics*, vol. 9, no. 12, p. 2152, 2020.

[8] A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, “Intrusion Detection System for Internet of Things Based on Temporal Convolution Neural Network and Efficient Feature Engineering,” *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–16, Dec. 2020, doi: 10.1155/2020/6689134.

[9] N. Hussain and P. Rani, “Comparative studied based on attack resilient and efficient protocol with intrusion detection system based on deep neural network for vehicular system security,” in *Distributed Artificial Intelligence*, CRC Press, 2020, pp. 217–236.

[10] G. Rathee, N. Jaglan, S. Garg, B. J. Choi, and K.-K. R. Choo, “A Secure Spectrum Handoff Mechanism in Cognitive Radio Networks,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 959–969, Sep. 2020, doi: 10.1109/TCCN.2020.2971703.

[11] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, “Machine Learning and Deep Learning Approaches for CyberSecurity: A Review,” *IEEE Access*, vol. 10, pp. 19572–19585, 2022, doi: 10.1109/ACCESS.2022.3151248.

[12] P. Rani *et al.*, "Federated Learning-Based Misbehavior Detection for the 5G-Enabled Internet of Vehicles," *IEEE Trans. Consumer Electron.*, vol. 70, no. 2, pp. 4656–4664, May 2024, doi: 10.1109/TCE.2023.3328020.
[13] P. Rani and M. H. Falaah, "Real-Time Congestion Control and Load Optimization in Cloud-MANETs Using Predictive Algorithms," *NJF Intelligent Engineering Journal*, vol. 1, no. 1, pp. 66–76, 2024.
[14] A. Marzano *et al.*, "The Evolution of Bashlite and Mirai IoT Botnets," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, Natal: IEEE, Jun. 2018, pp. 00813–00818. doi: 10.1109/ISCC.2018.8538636.
[15] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1–11, Aug. 2018, doi: 10.1016/j.jisa.2018.05.002.
[16] A. Singh *et al.*, "Smart Traffic Monitoring Through Real-Time Moving Vehicle Detection Using Deep Learning via Aerial Images for Consumer Application," *IEEE Trans. Consumer Electron.*, vol. 70, no. 4, pp. 7302–7309, Nov. 2024, doi: 10.1109/TCE.2024.3445728.
[17] M. Al-Hawawreh, E. Sitnikova, and F. Den Hartog, "An Efficient Intrusion Detection Model for Edge System in Brownfield Industrial Internet of Things," in *Proceedings of the 3rd International Conference on Big Data and Internet of Things*, Melbourn VIC Australia: ACM, Aug. 2019, pp. 83–87. doi: 10.1145/3361758.3361762.
[18] S. Latif, Z. Idrees, Z. Zou, and J. Ahmad, "DRaNN: A Deep Random Neural Network Model for Intrusion Detection in Industrial IoT," in *2020 International Conference on UK-China Emerging Technologies (UCET)*, Glasgow, United Kingdom: IEEE, Aug. 2020, pp. 1–4. doi: 10.1109/UCET51115.2020.9205361.
[19] Y. Li *et al.*, "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," *Measurement*, vol. 154, p. 107450, Mar. 2020, doi: 10.1016/j.measurement.2019.107450.
[20] B. Bhola *et al.*, "Quality-enabled decentralized dynamic IoT platform with scalable resources integration," *IET Communications*, 2022.
[21] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.
[22] J. Ahmad, S. A. Shah, S. Latif, F. Ahmed, Z. Zou, and N. Pitropakis, "DRaNN_PSO: A deep random neural network with particle swarm optimization for intrusion detection in the industrial internet of things," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 8112–8121, Nov. 2022, doi: 10.1016/j.jksuci.2022.07.023.
[23] A. Dawoud, S. Shahristani, and C. Raun, "Deep learning and software-defined networks: Towards secure IoT architecture," *Internet of Things*, vol. 3–4, pp. 82–89, Oct. 2018, doi: 10.1016/j.iot.2018.09.003.
[24] W. G. Negera, F. Schwenker, T. G. Debelee, H. M. Melaku, and Y. M. Ayano, "Review of botnet attack detection in SDN-enabled IoT Using machine learning," *Sensors*, vol. 22, no. 24, p. 9837, 2022.
[25] P. Krishnan, J. S. Najeem, and K. Achuthan, "SDN Framework for Securing IoT Networks," in *Ubiquitous Communications and Network Computing*, vol. 218, N. Kumar and A. Thakre, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 218. , Cham: Springer International Publishing, 2018, pp. 116–129. doi: 10.1007/978-3-319-73423-1_11.
[26] I. Indre and C. Lemnaru, "Detection and prevention system against cyber attacks and botnet malware for information systems and Internet of Things," in *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania: IEEE, Sep. 2016, pp. 175–182. doi: 10.1109/ICCP.2016.7737142.
[27] T. Hasan, A. Akhunzada, T. Giannetsos, and J. Malik, "Orchestrating SDN Control Plane towards Enhanced IoT Security," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, Ghent, Belgium: IEEE, Jun. 2020, pp. 457–464. doi: 10.1109/NetSoft48620.2020.9165424.
[28] N. Kumar Agrawal *et al.*, "TFL-IHOA: Three-Layer Federated Learning-Based Intelligent Hybrid Optimization Algorithm for Internet of Vehicle," *IEEE Trans. Consumer Electron.*, vol. 70, no. 3, pp. 5818–5828, Aug. 2024, doi: 10.1109/TCE.2023.3344129.
[29] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *Ieee Potentials*, vol. 13, no. 4, pp. 27–31, 1994.
[30] S. K. Satapathy, A. K. Bhoi, D. Loganathan, B. Khandelwal, and P. Barsocchi, "Machine learning with ensemble stacking model for automated sleep staging using dual-channel EEG signal," *Biomedical Signal Processing and Control*, vol. 69, p. 102898, 2021.
[31] J. B. Awotunde, C. Chakraborty, and A. E. Adeniyi, "Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, p. 7154587, Jan. 2021, doi: 10.1155/2021/7154587.
[32] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, p. 2852, 2021.

[33] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, Ieee, 2009, pp. 1–6.
[34] G. Meena and R. R. Choudhary, "A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA," in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, IEEE, 2017, pp. 553–558.
[35] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, IEEE, 2015, pp. 1–6.