International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

Deep Learning Based Network Intrusion Detection System using CNN-LSTM

Aatray Merothia[1], Prabhat Kumar Ray [2], Namrata Dhanda [3], Bommuraj[4,]
[1,2,3,4]Department of Computer Science and Engineering, Amity School of Engineering and Technology Lucknow, India
Emails: *jskfkgkratos@gmail.com [1], prabhatkumarrai62011@gmail.com [2], ndhanda510@gmail.com [3], amisra@lko.amity.edu [4]*

Abstract
Abstract— The increasing complexity of cyber threats has made traditional security systems insufficient for protecting modern network infrastructures. Advanced persistent threats, botnets, and multi-stage cyber attacks require intelligent systems capable of analyzing large volumes of network traffic. Deep learning techniques provide an effective solution by automatically learning complex patterns from data without extensive manual feature engineering. By combining CNN and LSTM architectures, the proposed system can analyze both structural and sequential characteristics of network traffic, enabling more accurate detection of malicious activities. This approach enhances the capability of intrusion detection systems and contributes to improving the overall reliability and security of modern computer networks. These days, with everyone glued to their devices and the internet running pretty much everything, network security has become a major issue for both businesses and regular folks. Cyber-attacks — DDoS, malware, phishing, people trying to sneak into accounts — they aren't just happening more often, they're getting trickier too. The problem is, traditional Intrusion Detection Systems aren't keeping up. Systems that rely on fixed signatures or rules struggle with new, unknown attacks and end up flagging way too many false positives, which honestly just becomes a headache. So, this research focuses on a new approach: a deep learning-based Network Intrusion Detection System that uses a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The idea is simple. The CNN looks through raw network traffic data and pulls out meaningful features automatically. Then, the LSTM checks for patterns over time, spotting suspicious behavior as it unfolds. By combining CNN and LSTM, the system digs into both what's happening at a single moment and how things change over time, letting it catch malicious activity more accurately. To build and test the model, using well-known datasets like NSL-KDD and CICIDS2017. These datasets cover all sorts of attacks: DoS, Probe, Remote-to-Local, User-to-Root — you name it. Before jumping into training, they cleaned up the data, normalized it, picked out the most useful features, and converted everything into numbers so the deep learning algorithms could make sense of it. These steps help the model focus and boost its performance.To see how well the model works, we measured accuracy, precision, recall, and F1-score. Turns out, the hybrid CNN-LSTM system outperformed traditional machine learning methods. It nailed higher detection accuracy and cut down on false positives, making network security a little less stressful and a lot more reliable. The proposed deep learning-based intrusion detection system can provide an effective solution for enhancing network security in modern environments. It can be applied to enterprise networks, cloud infrastructures, and large-scale communication systems to detect malicious activities in real time. Future research can focus on optimizing the model for real-time deployment, integrating attention mechanisms, and extending the system for Internet of Things (IoT) network security.

Keywords—*Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Network Intrusion Detection System (IDS), Deep Learning.*

1. Introduction
The explosion of internet technology has changed the way we live and work, but it's also opened the door

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

to all kinds of security problems. These days, people and businesses depend on computer networks for almost everything—sharing data, running online services, moving money around, you name it. But the more we rely on these networks, the bigger the target they become. Cyber attackers continuously evolve their techniques to bypass traditional security mechanisms, making it necessary to develop advanced detection systems that can adapt to new threats. Deep learning models have gained popularity in cybersecurity because they can process large datasets and identify hidden patterns within complex network environments. Integrating deep learning into intrusion detection systems can help organizations improve threat detection capabilities and protect critical network resources from unauthorized access and attacks. Cyber threats like malware, DDoS attacks, phishing scams, and hackers trying to sneak in are everywhere. When these attacks hit, they can leak sensitive data, shut down services, and rack up huge financial losses. So, keeping networks safe from these threats isn't just important—it's absolutely necessary. That's where a Network Intrusion Detection System, or IDS, steps in. Think of it as a security guard for network traffic, always on the lookout for anything suspicious or dangerous. Traditionally, IDS use two main tricks: signature-based detection and anomaly-based detection. The first one matches network activity to a list of known attack patterns. It works well for old threats, but misses anything new. Anomaly-based detection, on the other hand, tries to flag anything that looks out of the ordinary. The catch? It tends to ring the alarm bell a little too often, even when nothing's wrong. To get around these issues, researchers have started turning to machine learning and deep learning. These approaches don't just follow rules—they learn from huge amounts of data, picking up on patterns people might miss. Deep learning, in particular, is great for this. Convolutional Neural Networks (CNNs) are awesome at pulling out useful features from raw data, and Long Short-Term Memory (LSTM) networks are built to spot patterns that unfold over time. In this research, we're putting both to work. By building a hybrid system that combines the strengths of CNNs and LSTMs, we can spot attacks more accurately than before. The system digs into network traffic, picks out anything shady, and helps keep networks safer overall.

2. Literature Review

Several studies have explored the use of deep learning techniques in network intrusion detection [1]. Researchers have applied models such as Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), and Autoencoders to identify malicious network behavior. CNN-based intrusion detection systems have demonstrated strong performance in feature extraction tasks, while LSTM-based models have shown effectiveness in analyzing sequential network traffic patterns. Hybrid models combining multiple deep learning techniques have been proposed to improve detection accuracy and reduce false positives. These studies highlight the potential of deep learning methods to enhance the performance of modern intrusion detection systems. Network Intrusion Detection Systems, or IDS, have long been a go-to for keeping computer networks safe from cyber-attacks. In the early days, these systems mostly leaned on signature-based and rule-based methods. Basically, they'd scan network traffic and look for known attack patterns saved in a big database. If something matched, they'd flag it. The catch? Signature-based IDS are great at catching threats you already know about, but anything new slips right past them. Plus, someone has to keep updating that database all the time. To get around these issues, researchers started using machine learning for intrusion detection. Algorithms like Support Vector Machines, Decision Trees, Random Forest, and k-Nearest Neighbors (KNN) all came into play. They sort network traffic as either normal or malicious by learning from past data. This definitely helped, since the systems could spot patterns on their own. But there's still a downside. Traditional machine learning needs people to handpick features before the algorithm can do its job, and as network data gets more complicated, that job gets harder. That's where deep learning comes in. Lately, it's

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

been getting a lot of attention in network security. Deep learning models can sift through massive datasets and learn complex patterns on their own, no manual feature-picking required. Convolutional Neural Networks (CNNs) are especially good at pulling out useful features from network traffic and spotting the subtle signs of an attack. On the other hand, Long Short-Term Memory (LSTM) networks are great at picking up on how traffic changes over time, which helps catch attacks that play out in sequences [1]. Some researchers have even started combining CNNs and LSTMs to build hybrid models for intrusion detection. Here, CNNs handle the feature extraction, while LSTMs make sense of the sequence and timing in the data. These hybrid models have been shown to boost detection accuracy and cut down on false alarms compared to the older machine learning methods. So, blending CNN and LSTM really does give us a smarter and more dependable way to spot cyber threats as they happen.

3. Problem Statement

Cyber-attacks keep getting smarter and more common as computer networks and online services grow. Most traditional Network Intrusion Detection Systems (IDS) work by looking for known attack signatures in network traffic. They're solid at catching threats people have already seen, but they usually miss new or unknown attacks. On top of that, a lot of these older IDS setups flag too many harmless activities as threats and just can't handle the huge, messy flood of modern network data. Switching to machine learning has helped—detection rates have gone up. Still, these systems lean a lot on people manually picking out which features matter, and they can get bogged down when the data gets too complex or high-dimensional. So, there's a real need for something smarter and more automated. A system that actually learns the tough patterns in network traffic and nails down what's malicious, even as attacks keep evolving. That's the goal of this research: building a deep learning intrusion detection system using a hybrid CNN-LSTM model. The idea is to boost detection accuracy and, ultimately, make networks safer. Another challenge faced by traditional intrusion detection systems is their limited ability to adapt to dynamic network environments. Modern networks generate large volumes of diverse traffic, making it difficult for rule-based systems to analyze and classify data effectively. Additionally, many existing detection systems struggle to maintain high detection accuracy while minimizing false alarms. This research addresses these challenges by developing a hybrid deep learning model capable of learning complex patterns in network traffic and improving the detection of both known and unknown attacks.

4. Proposed Methodology

This system sets out to build a deep learning Network Intrusion Detection System using a combination of CNN and LSTM layers. The goal? Spotting malicious activity in network traffic. Here's how it works, step by step. First, you start with a public dataset like NSL-KDD or CICIDS2017. These datasets have a mix of normal network activity and all sorts of cyberattacks. Before diving into modeling, you have to clean the data [2]. That means getting rid of missing or inconsistent entries, turning categorical data into numbers, and normalizing everything so the model can actually learn something useful. After that, you split the dataset into training and testing sets. After preprocessing, the processed data is fed into the CNN-LSTM deep learning model. The Convolutional Neural Network (CNN) component is responsible for automatically extracting important features from the network traffic data. These extracted features are then passed to the Long Short-Term Memory (LSTM) layer, which analyzes sequential patterns and temporal dependencies in the traffic data. At the end, the model runs its results through some fully connected layers and a softmax classifier, which sorts the network traffic as either normal or malicious. To see how well everything works, you use metrics like accuracy, precision, recall, and F1-score. This hybrid setup boosts detection power and makes the intrusion detection system much more effective overall. The proposed methodology focuses on integrating deep learning techniques into the intrusion detection process to

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

enhance system performance. By combining CNN and LSTM layers, the model is capable of learning hierarchical feature representations and temporal relationships in the data. The CNN layers perform convolution operations to detect important features, while the LSTM layers analyze time-based sequences in network traffic flows. This hybrid approach allows the system to detect multi-stage cyber-attacks that may occur over a period of time.

5. System Architecture and Flowchart

5.1.System Architecture

This Network Intrusion Detection System uses a mix of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to spot suspicious activity in network traffic. Here's how it works: everything starts with collecting traffic data from well-known datasets like NSL-KDD or CICIDS2017. Shown in Figure 1.

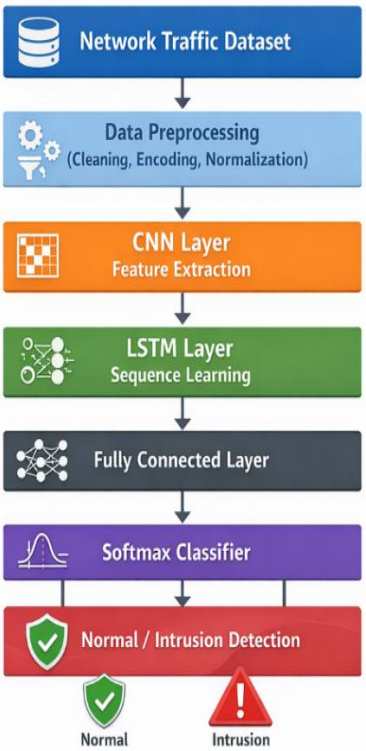

Figure 1 Hybrid deep learning NIDS architecture

These datasets hold both regular network activity and a variety of cyber-attacks. Once the data's in, it goes through preprocessing. That means cleaning it up, normalizing values, and turning any categories into numbers the model can understand. This step gets the data ready for the deep learning model. Next, the system sends data through the CNN layer. The CNN digs through the network traffic, pulling out the features and patterns that might signal trouble. After that, the features move on to the LSTM layer. This part focuses on how things change over time, picking up sequential or time-based patterns in the data. In the final stretch, the information flows into fully connected layers and then through a SoftMax classifier. This last bit sorts the network traffic into either normal or intrusion. By combining CNN and LSTM, the system picks up on even complex attack patterns and pushes the accuracy of intrusion detection higher. Shown in Figure 2.

5.2.Flowchart of The System

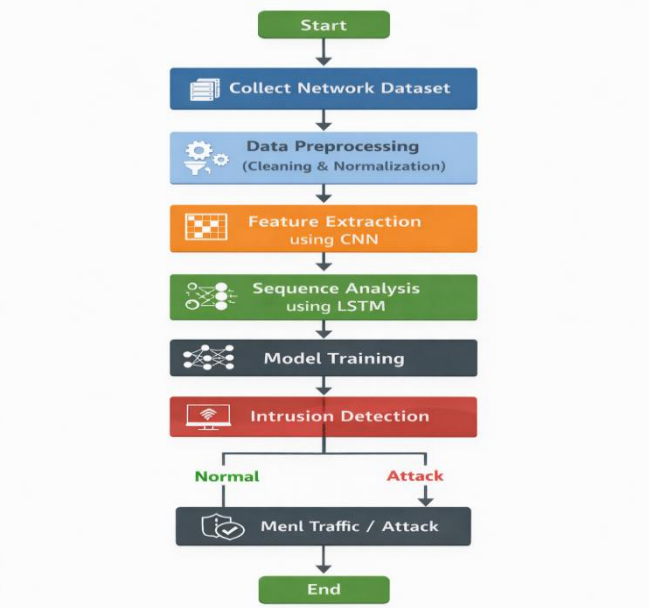

Figure 2 Network intrusion detection process flowchart

Here's how the proposed Network Intrusion Detection System works, step by step. First, you grab a network traffic dataset[3]. This isn't just a list of ordinary activity — it's got records of what normal

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

use looks like, plus all sorts of cyber-attacks mixed in. That dataset kicks off the whole process. Next up is data preprocessing, think of this as cleaning house. You fill in missing values, turn any text categories into numbers, and normalize everything so the data's ready for the model. If you skip this, the model just won't learn properly. Once that's done, the system hands things over to a Convolutional Neural Network (CNN). This part digs through the traffic data and pulls out the features and patterns that matter — the stuff a human might miss. After the CNN does its job, the extracted features go to a Long Short-Term Memory (LSTM) layer [4]. The LSTM's specialty is spotting patterns over time. It looks for sequences and trends — basically, it pays attention to how attacks unfold, not just isolated events. Then comes the training and classification phase. The model learns from all that prepped data, figuring out how to tell the difference between normal activity and an attack. And finally, the system gets to work in the real world. It checks live network traffic and sorts it: normal, or threat. This whole setup helps catch cyber threats more accurately and keeps the network safer overall.

6. Hybrid Cnn–Lstm Model Architecture

This model for the Network Intrusion Detection System uses a mix of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The idea is pretty simple: CNNs pick up important patterns from the data, while LSTMs track how things change over time. Blending both helps the system spot intrusions more accurately shown in Figure 3

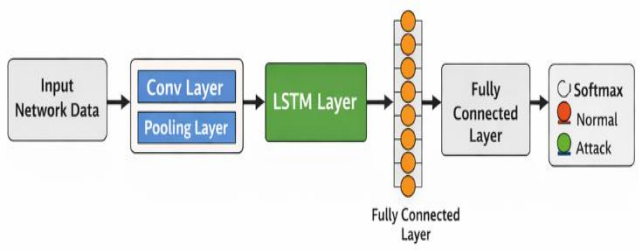

Figure 3 CNN-LSTM Model Architecture

- It all starts at the input layer. Preprocessed network traffic goes in, things like protocol type, packet size, duration, and other details that describe what's happening in the network.
- CNN layers go to work. They sift through the data, automatically pulling out features by applying filters that notice key relationships and patterns. Pooling layers usually come next. They shrink things down, keeping the most valuable information and dropping the extra noise.
- Then comes the LSTM layer. LSTMs are great with sequences, so they look for patterns and dependencies over time. That's important because some cyber-attacks play out in steps or have timing clues that aren't obvious in a single snapshot.
- After all this, you've got a set of features that move on to a dense, fully connected layer. This is where the model decides what it's looking at—sorting the traffic into categories.
- The final step uses a SoftMax layer, so the system can say if the traffic is normal or if it fits a particular kind of attack.

By bringing CNN and LSTM together, the model makes the most of both pattern recognition and temporal analysis. It's a stronger, smarter approach to spotting intrusions.

7. Data Preprocessing and Training

7.1.Dataset Analysis and Conclusion

Looking through the intrusion detection dataset, you start to see a few big patterns in how network traffic behaves and how cyber-attacks are spread out. There's a mix of everyday activity and plenty of different types of attacks: Denial of Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). Most of what's recorded is regular traffic, but some attack types barely show up. One thing jumps out right away: DoS attacks are way more common than the others. That tells you DoS is a pretty frequent threat in network environments [5]. At the same time, R2L and U2R attacks don't show up nearly as often. This creates a classic class imbalance, which can

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

mess with how well models learn to spot rarer attacks. Digging into the dataset, you notice certain network features stand out when comparing normal and malicious traffic. Stuff like how long a connection lasts, the number of packets sent, protocol type, and service usage—they all tend to shift noticeably when an attack is happening. These patterns are gold for machine learning and deep learning models. Models can use them to separate the routine traffic from the suspicious stuff. In short, the dataset makes it clear: there are attack patterns, some balancing issues between classes, and distinct relationships between network features. All of these shapes how you handle preprocessing, which features you focus on, and the kind of model training that will work best for intrusion detection [7-10].

7.2.Data Preprocessing

The performance of the proposed Network Intrusion Detection System is evaluated using a publicly available intrusion detection dataset such as NSL-KDD or CICIDS2017, these datasets are widely used in cybersecurity research for evaluating the effectiveness of intrusion detection models. The NSL-KDD dataset is widely used for evaluating intrusion detection systems and addresses limitations of the KDD Cup 1999 dataset. The datasets contain large volumes of network traffic records that include both normal network activities and various types of cyber-attacks.

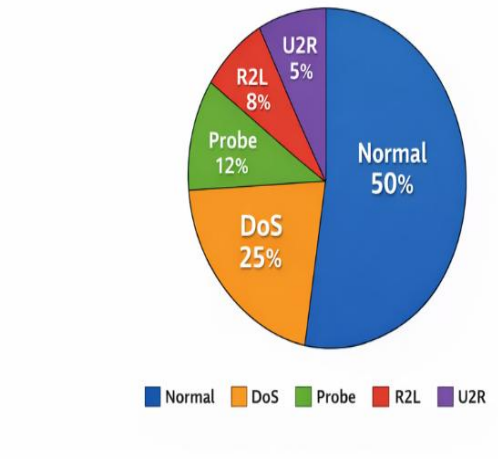

Figure 4 Dataset Distribution

Effective data preprocessing improves the quality of input data and ensures better training results. Feature selection techniques may also be applied to identify the most relevant attributes from the dataset. Removing irrelevant or redundant features helps reduce computational complexity and improves model efficiency. Additionally, balancing techniques such as oversampling or under sampling may be used to address class imbalance issues in intrusion detection datasets. It is one of the most important parts of building a good Network Intrusion Detection System. Raw network data is messy—lots of missing values, extra features you don't need, and categories that deep learning models just can't read as they are. Cleaning this up isn't just busy work. It really boosts how well the model works. For this project, we are using a public data set like NSL-KDD or CICIDS2017. The first thing we do is clean the data. That means scrubbing out anything incomplete records with missing info, duplicates, you name it. This way, we know what's left is solid and ready for training. Next up, there are those categorical features: things like protocol type, service, and flag. Deep learning models only work with numbers, so we convert those categories into numerical values with label encoding or one-hot encoding. It's not optional, it's just how these models work. Once everything's in numbers, we normalize or scale the features. This puts all the values in a similar range, which helps the model learn faster and stops any one feature from overpowering the rest. After all that, we split the data—usually 80% for training and 20% for testing.

7.3.Model Training

The training set goes straight into the CNN-LSTM model, and the testing set waits to see how well everything turned out. Now, about that model. We feed the clean, prepped network data into the CNN-LSTM. First, the Convolutional Neural Network (CNN) layers jump in and automatically pull out the most important features and patterns buried in the traffic data. It's like having a sharp filter for what matters. Then, the Long Short-Term Memory (LSTM) layer takes over. This part digs into sequences and time-based patterns—basically, it

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

watches how things change over time to spot attacks that don't show up all at once. During model training, the network parameters are optimized through backpropagation and gradient descent algorithms. The training process adjusts the weights of the neural network to minimize the loss function and improve prediction accuracy. Hyperparameters such as learning rate, batch size, and number of epochs play an important role in determining the performance of the deep learning model. Proper tuning of these parameters ensures that the model learns effectively without overfitting. Model optimization is an important step in improving the performance and efficiency of the proposed CNN-LSTM based intrusion detection system. During the training process, several hyperparameters are adjusted to ensure that the model learns effectively and produces accurate predictions. One of the key optimization techniques used in this research is the Adam optimizer, which helps update the model weights efficiently during training by minimizing the loss function. The Adam optimizer combines the advantages of adaptive learning rates and momentum-based gradient descent, making it suitable for deep learning applications. The model also uses activation functions such as Rectified Linear Unit (ReLU) in the hidden layers and SoftMax in the output layer. ReLU helps the network learn complex patterns by introducing non-linearity, while the SoftMax function converts the model output into probability values for classification. Other important parameters that influence model performance include batch size, learning rate, and number of epochs. These parameters are carefully selected to balance training time and model accuracy. A suitable batch size ensures efficient computation, while the learning rate controls how quickly the model updates its parameters during training. Additionally, techniques such as regularization and dropout layers may be used to prevent overfitting and improve the generalization capability of the model. By optimizing these parameters and techniques, the CNN-LSTM model can achieve higher detection accuracy and improved intrusion detection performance. We run this training over several epochs and set a batch size to make sure the learning sticks. When training wraps up, we test the model on the set we kept aside. This shows me how well it can tell the difference between normal network traffic and potential attacks.

8. Evaluation and Results

8.1.Model Evaluation

To see how well the CNN-LSTM Network Intrusion Detection System works, we looked at a few common metrics. Basically, these numbers show how good the model is at spotting normal traffic versus malicious activity. The big ones we focused on were accuracy, precision, recall, and F1-score.

- Accuracy is straightforward, it tells you what percentage of the traffic the model got right overall. If you see high accuracy, that means the system isn't missing much.
- Precision is a bit more specific. It looks at all the time the model raised the alarm for an intrusion and checked how many of those were real threats. High precision means you're not getting bombarded with false alarms.
- Recall, or detection rate, is about how many real attacks the system catches. If recall is high, it's catching most of the threats that come its way.
- F1-score balances precision and recall. It's a single number that gives you a sense of how well the model does when things are a bit messy, like if there's way more normal traffic than attacks.
- We also used a confusion matrix to lay out the results visually showing how many attacks and normal cases the model got right or wrong. Altogether, these tools give a pretty clear picture of how effective the system is at spotting network intrusions.

8.2.Results

We tested the hybrid CNN–LSTM Network Intrusion Detection System using well-known datasets like NSL-KDD and CICIDS2017. These datasets include all kinds of network traffic—some of it normal, some of it packed with different types of cyber-attacks.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

First, we cleaned up the data, normalized everything, and converted any categorical features so the model could understand them. After that, we split the data into training and testing sets to see how well the model would really perform. For training, we ran the CNN-LSTM model through several epochs, tweaking the learning rate and batch size to get the best results. The CNN layers dug into the network traffic data, pulling out useful features, while the LSTM layer tracked patterns over time and caught the sequence in the data. Once the training was done, we put the model to the test on new, unseen data—checking if it could accurately sort out normal traffic from malicious activity. We measured performance using metrics like accuracy, precision, recall, and F1-score. The results? The CNN-LSTM model scored high on detection accuracy and did a better job at classifying traffic than older machine learning methods. It really stood out when it came to catching complex attacks and keeping false positives down. In the end, all this testing shows that the CNN-LSTM intrusion detection system isn't just reliable, it's efficient, and it seriously boosts network security by spotting cyber-attacks when they happen. The experimental analysis demonstrates that the hybrid CNN-LSTM model performs better in detecting complex attack patterns compared to traditional machine learning algorithms. Graphical representations such as confusion matrices and performance charts can be used to visualize the classification results. These visual tools help evaluate how well the model distinguishes between normal network traffic and various attack categories.

9. Advantages of The Proposed System

This deep learning Network Intrusion Detection System, built with a hybrid CNN–LSTM setup, does a lot that traditional systems just can't keep up with. For starters, it's way more accurate. The old methods depend on a bunch of fixed rules or signatures, so they miss a lot of the more complicated attacks. The CNN–LSTM model sidesteps that by learning patterns straight from massive piles of network traffic data. That means it's better at spotting sneaky attacks. Another big win here is that it handles feature extraction on its own. In the past, people had to spend hours—or even days—handpicking which features mattered for detection, and honestly, important stuff still slipped through the cracks. With this model, the CNN part digs through the data and pulls out what matters, saving time and catching things humans might miss. And the system isn't just stuck with what's already known. Signature-based systems can only cause flag attacks they've seen before, so anything new slips right by. But the CNN–LSTM system learns how normal network traffic behaves, so when something weird pops up, it can spot it—even if it's never seen that attack before. It also does a much better job of lowering false alarms. CNN looks at the structure of the data, LSTM handles patterns over time, and together they separate harmless activity from real threats. The result? You get an intrusion detection system you can trust. Plus, this model faces no problem handling the huge volumes of data you get in real networks these days. Whether you're running an enterprise, working in the cloud, or dealing with IoT devices, it keeps up with the traffic and doesn't get bogged down. In short, it's built for the real world.

10. Discussion

The experimental results obtained from the proposed CNN–LSTM based Network Intrusion Detection System demonstrate the effectiveness of deep learning techniques in identifying malicious network activities. The hybrid architecture combines the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to analyze both spatial and temporal patterns present in network traffic data. This combination enables the system to capture complex relationships within the dataset and significantly improves intrusion detection performance. The CNN component of the model plays a crucial role in automatically extracting meaningful features from the network traffic dataset. Unlike traditional machine learning methods that rely heavily on manual feature engineering, the CNN layers are capable of learning important characteristics directly from the raw input data. This not only reduces the complexity of the preprocessing

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

stage but also allows the system to identify hidden patterns that may not be easily captured through manual methods. The LSTM layer further enhances the performance of the system by analyzing sequential dependencies and temporal behaviors within network traffic flows. Many cyber-attacks occur over a sequence of events rather than a single instance of abnormal activity. By capturing these time-based relationships, the LSTM model improves the system's ability to detect sophisticated and evolving attack patterns. The evaluation results indicate that the proposed hybrid model achieves higher accuracy and better classification performance compared to many traditional intrusion detection techniques. The model also demonstrates improved precision and recall values, which indicates that it can effectively detect attacks while minimizing false alarms. A lower false positive rate is particularly important in real-world environments because excessive alerts can reduce the efficiency of network administrators and security systems. The results highlight the importance of combining multiple deep learning techniques for intrusion detection tasks. Hybrid architectures allow models to capture different aspects of network traffic behavior, improving their ability to detect cyber threats. The findings also suggest that deep learning models can significantly reduce the reliance on manual feature engineering while improving detection accuracy. Despite these advantages, there are still certain challenges associated with the proposed approach. Deep learning models generally require large datasets and significant computational resources for training. Additionally, the training process may take longer compared to simpler machine learning models. However, with the continuous advancement of computing technologies and availability of large cybersecurity datasets, these challenges are gradually becoming less significant. Overall, the results show that the CNN–LSTM based intrusion detection system provides a promising solution for improving network security and detecting modern cyber threats more effectively.

Conclusions

In this research, a deep learning-based Network Intrusion Detection System using a hybrid CNN–LSTM architecture was proposed to improve the detection of malicious network activities. With the increasing number of cyber-attacks and the limitations of traditional intrusion detection systems, there is a strong need for intelligent and automated security solutions. The proposed system combines the feature extraction capability of Convolutional Neural Networks with the sequence learning ability of Long Short-Term Memory networks to analyze network traffic data effectively. The model was trained and evaluated using a standard intrusion detection dataset after performing necessary preprocessing steps such as data cleaning, encoding, and normalization. Experimental results demonstrate that the hybrid CNN–LSTM model is capable of accurately classifying network traffic as normal or malicious. The system shows improved performance in terms of detection accuracy, precision, recall, and reduced false positive rates compared to traditional machine learning approaches. Overall, the proposed approach provides an effective solution for detecting cyber threats and enhancing network security and that deep learning models can play a crucial role in strengthening cybersecurity systems. By leveraging advanced neural network architectures, intrusion detection systems can better adapt to evolving cyber threats and provide more reliable protection for network infrastructures. Future improvements in deep learning techniques and computing power are expected to further enhance the effectiveness of such systems. The integration of deep learning techniques into intrusion detection systems can significantly improve their ability to identify complex and evolving attack patterns in modern network environments.

Acknowledgement

The successful completion of this research would not have been possible without the support and guidance of several individuals and institutions, to whom we wish to express our deepest gratitude. First and foremost, we extend our sincere thanks to our supervisor, Dr. Namrata Dhanda and Dr. Anuradha

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2255-2264
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0303

Misra. Their invaluable guidance, insightful feedback, and unwavering encouragement throughout the course of this work were instrumental in shaping the direction and quality of this research.
We would like to acknowledge the authors of the foundational studies and the creators of the open-source libraries, particularly the Hugging Face transformers team, upon whose shoulders this work stands. Their contributions to the scientific community have been indispensable.

References

[1] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," IEEE Access, vol. 5, pp. 221–231, 2017.
[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[4] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[5] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6.
[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. International Conference on Information Systems Security and Privacy, 2018, pp. 108–116.
[7] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," Expert Systems with Applications, vol. 41, no. 4, pp. 1690–1700, 2014.
[8] M. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," IEEE Access, vol. 5, pp. 2194–2208, 2017.
[9] K. Kim, Y. Kim, H. Kim, and J. Kim, "LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," IEEE Access, vol. 4, pp. 1–10, 2016.
[10] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. Military Communications and Information Systems Conference, 2015, pp. 1–6.