International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Future-Driven Network Anomaly Detection: Transitioning from Machine Learning to Deep Learning Models

Mrs. J. Gayathri., M.E., (Ph.D).,[1]
Department of Artificial Intelligence and Data Science Gnanamani College of Technology, Namakkal, Tamil Nadu, India – 637018.

Mr. G. Sivakumar., M E., (Ph.D)., [2]
Department of Artificial Intelligence and Data Science Gnanamani College of Technology, Namakkal, Tamil Nadu, India - 637018.

Mr. F. Cyril Mervyn [3]
Department of Artificial Intelligence and Data Science Gnanamani College of Technology, Namakkal, Tamil Nadu, India - 637018.

Mr. S. S. Irfaan [4]
Department of Artificial Intelligence and Data Science Gnanamani College of Technology, Namakkal, Tamil Nadu, India – 637018.

Mr. S. Gowtham [5]
Department of Artificial Intelligence and Data Science Gnanamani College of Tec hnology, Namakkal, Tamil Nadu, India - 637018.

Mr S. Nithesh [6]
Department of Artificial Intelligence and Data Science Gnanamani College of Technology, Namakkal, Tamil Nadu, India – 637018.

Abstract - Out here, digital tools are spreading fast. Cloud systems keep growing too. Devices link together more every day. Because of that, the flow through networks gets heavier. It also gets harder to follow. All this change opens doors for many kinds of online attacks. Spotting odd behavior in huge streams of data? Not easy anymore. Especially when threats shift shape over time. Strange signs hide better now. Most older security tools spot intrusions using fixed rules, working just against familiar breaches. Instead of relying on set templates, some smarter methods study past events to catch suspicious behavior, yet struggle when data gets too messy or shifts suddenly. What changes the game is deep neural networks - they grasp tangled connections without needing step-by-step guidance. Out there, a live network monitoring tool mixes smart algorithms with layered learning tricks. Built on the CICIDS-2017 collection of traffic samples, it learns what normal looks like. Instead of just one method, it tries two - starting with a forest of decision paths set as reference. Then comes a brain-like grid, shaped like a net, tuned to catch oddities faster. Behind the scenes, an engine built with FastAPI feeds fake flows into the loop. These guesses pop up instantly on a screen that never sleeps. Deep learning boosts detection power when it comes to tricky, hidden threats. What stands out is how well these models adjust and perform under pressure. Moving forward, relying on such methods makes sense for stronger digital defenses. Systems grow smarter, not just bigger, by using this approach

OBJECTIVES

This project aims to build a smart way to spot unusual patterns in data flow across networks. What happens next depends on how well the system learns what normal looks like. Instead of guessing, it watches

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

closely then flags anything off rhythm. Patterns shift over time, so the method must adapt without constant oversight. Efficiency matters because delays can hide problems. A quiet signal might mean trouble just as loud bursts do. Learning from real examples helps avoid false alarms later down the line.

The specific objectives are:

- To analyze network traffic and identify patterns of normal and abnormal behavior
- To preprocess and clean network data for effective model training
- To implement Random Forest as a baseline machine learning model
- To develop a CNN model for deep learning-based anomaly detection
- To compare the performance of machine learning and deep learning models
- Watch network actions as they happen
- Fine-tuning the system sharpens results without triggering unnecessary alerts

INTRODUCTION

Nowadays, more people use internet tools every day - this means networks carry far heavier loads than before. Companies depend on these connections to share messages, keep files safe, store information, also run daily tasks. Because so much rides on stable links, protecting them from threats matters like never before.

Out of nowhere, cyberattacks grow sharper, slipping past old-school defenses. Because they change so fast, spotting them gets tricky without smarter tools. DDoS strikes, sneaky port scans, or hidden botnets - all these mess with operations while stealing private info. Signature-based guards fall short since they only recognize what's already been seen. When threats morph, those fixed rules just can't keep up.

Most of the time, machine learning helps spot oddities by studying old examples. Still, such systems usually need hand-crafted inputs. On top of that, they can stumble when dealing with messy, layered information.

Learning deeply lets computers spot trends on their own. What makes CNNs stand out is how they untangle tangled feature links. Shifts in network flow? They adjust without needing rewrites. Odd behaviors slip through less often under their watch.

From the start, this work builds a tool to spot odd network activity as it happens, combining smart algorithms from machine learning and deeper neural networks. Right away, spotting strange patterns becomes possible - then comes checking which method works better through live updates on a display that shows every shift.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Feature Category	Features Included	Description
Basic Network Attributes	Src IP dec, Src Port, Dst IP dec, Dst Port, Protocol,Flow Duration	Source and destination addresses and ports, protocol type, total flow duration
Packet Statistics	Total Fwd Packet, Total Bwd packets, Total Length of Fwd Packet, Total Length of Bwd Packet, Fwd/Bwd Packet Length Max/Min/Mean /Std	Counts and length stats of packets in forward and backward directions
Flow Metrics	Flow Bytes/s, Flow Packets/s, Flow IAT Mean/Std/Max/ Min, Fwd/Bwd IAT Total/Mean/Std /Max/Min	Flow-level bytes and packets per second, inter-arrival times (IAT) statistics
TCP Flags	FIN,SYN, RST, PSH, ACK, URG, CWR, ECE Flags (forward and backward)	Counts of TCP flag occurrences in both directions
Header Lengths	Fwd Header Length, Bwd Header Length	Lengths of packet headers in both directions
Packet Timing	Fwd Packets/s, Bwd Packets/s, Packet Length Min/Max/Mean /Std/Variance, Active Mean/Std/Max/ Min, Idle Mean/Std/Max/ Min	Packet rates, sizes, and active/idle time statistics
Bulk Features	Fwd Bytes/Bulk Avg, Fwd Packet/Bulk Avg, Fwd Bulk Rate Avg, Bwd Bytes/Bulk Avg, Bwd Packet/Bulk Avg, Bwd Bulk Rate Avg	Metrics describing bulk packet transmission characteristics
Subflow Statistics	Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes	Counts and sizes of subflows in both directions
Window and Segment Sizes	FWD Init Win Bytes, Bwd Init Win Bytes, Fwd Seg Size Avg, Bwd Seg Size Avg, Fwd Seg Size Min	Initial window sizes and segment size statistics
CMP Attributes	ICMP Code, ICMP Type	Internet Control Message Protocol codes and types

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Other Metrics	Down/Up Ratio, Average Packet Size, Total TCP Flow Time	Ratios and summary metrics of packet sizes and flow durations

METHODOLOGY

A step-by-step approach guides this project, beginning with cleaning up raw information before moving into building predictive systems. Following that phase comes live testing under dynamic conditions, where outcomes unfold as events progress. Each stage connects to the next, shaped by earlier results yet open to adjustments when needed.

To begin, studying network traffic helps spot what looks typical versus unusual. Realistic examples come from the CICIDS-2017 collection of activity records.

Missing pieces, repeated entries, and errors get cleared out first. Instead of categories, labels turn into two groups: Normal or Attack. To balance things, numbers go through a normalizing step using StandardScaler. Cleaning up comes before any labeling happens. Scaling adjusts each feature so everything lines up fairly. Before sorting anything, messy data must be fixed. Binary labels replace complex ones early on. StandardScaler steps in once the set looks clean. No gaps or glitches stay past the start. Normalization runs after duplicates take their exit.

Afterward, the data divides into two parts - one for training, another for testing. This time, a Random Forest steps in as the starting point for classification. Instead of relying on just one tree, it builds many, each adding slight improvements while keeping errors in check.

A pattern-finding system builds understanding through layered processing. As it learns, adjustments happen inside each level - shaping how details are noticed. Improvement comes not just from data but from repeated refinement over time.

Right away, a FastAPI backend kicks in, mimicking actual usage by pushing constant network flow. This live stream of information moves straight into processing, routing without delay to each model. As guesses form, they travel instantly to the display through WebSocket links. From there, updates appear as fast as they're made.

SYSTEM ARCHITECTURE

A web of parts links up inside the setup, each piece playing its role in spotting odd traffic patterns across the network. Module by module, they pass signals along, staying in step without needing constant direction. One shift in flow gets noticed fast because attention spreads through shared alerts. When something runs off track, the response comes from how pieces fit, not one single boss part calling shots.

1. Data Input Module
2. From simulations or datasets, it pulls details about network activity. Traffic information gets gathered through these sources instead of live systems.
3. Preprocessing Module
4. Cleans the data by removing missing values and duplicates.
5. Feature Scaling Module
6. Normalizes the data using StandardScaler.
7. Machine Learning Module
8. Implements Random Forest for baselineprediction.
9. Deep Learning Module
10. A twist on pattern spotting powers unusual behavior alerts. Layered filters scan data like a detective hunting odd clues. Sudden shifts get flagged through clever grid-based learning.
11. Evaluation Module
12. Performance of models gets checked through specific measurement tools. How well each one works shows up clearly when judged by these standards.
13. Dashboard Module

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

14. Displays real-time predictions and system status.
15. Streaming live updates happens through WebSocket connections, so tracking stays nonstop. Continuous oversight works because data flows instantly via WebSocket links.

RESULTS AND DISCUSSION

This study's test outcomes show machine learning works well for spotting odd behavior in network activity, just like deep learning does. Strong at catching common attack types, the Random Forest model served as the starting point. Built on combining many decision trees, it managed organized datasets without much trouble. Clear patterns led to steady outputs, thanks to how the method pools multiple models together.

Still, the Random Forest model showed its weak spots once situations got trickier - attacks that were too quiet or layered slipped through. Sometimes regular activity was flagged by mistake, seen as dangerous even though it wasn't. On top of that, sneaky intrusions barely stood out from everyday behavior, making them tough to catch.

Yet the Convolutional Neural Network handled tricky, hidden irregularities better. Because it learned key traits straight from raw data, spotting faint shifts in traffic flow came naturally. So when faced with sneaky intrusions or shifting network rhythms, its response stayed sharp. Though not perfect, its edge lay in reading between the lines of normal use.

Looking at the numbers made it clear. Higher precision and recall showed up in the CNN results when placed beside those from Random Forest, meaning fewer missed threats plus less noise. Balance mattered, which is why the F1-score tipped toward CNN too.

A key point stood out when looking at how the CNN adjusted over time. While the Random Forest sticks to fixed rules set early on, the CNN gets sharper with every new batch of information. Because of this trait, it handles shifting conditions better - especially where threats change shape gradually.

Imagine watching two minds work at once, side by side. That screen made contrasts pop, without needing to dig through numbers. One glance showed how each model reacted when things changed. It wasn't just about speed - how fast they spotted something mattered too. Confidence scores moved like heartbeats, revealing which guesses felt sure. Seeing all that unfold in real time shaped how we judged what actually worked.

It took a bit more power and time to run the CNN model, yet the jump in accuracy made up for it. When used out in the world, catching threats right matters far more than waiting an instant longer.

Deep learning models handle network anomalies better than older methods, showing greater consistency across different scenarios. Though both aim to spot unusual activity, one adapts more smoothly when conditions shift. Where conventional techniques struggle with complexity, neural networks adjust without constant reconfiguration. Their ability to learn patterns over time sets them apart in real-world applications.

CONCLUSION

Looking at how data moves through networks, This project shows how a live network threat detection setup was built using two types of smart algorithms. Instead of just one method, it used both traditional pattern recognition and layered neural networks. One approach learns from clear rules, while the other uncovers hidden patterns through repeated exposure. The test measured speed, accuracy, and response under pressure. Results showed that the deeper model adapted better when attacks changed form. Even though simpler methods kept up at first, they struggled as situations grew messy. Complex structures handled noise without losing track. What stood out was not raw power but flexibility in unclear moments. Deep layers adjusted where others stalled.

Starting off, the Random Forest model held up well when spotting familiar attack types. Still, it struggled whenever odd or intricate irregularities appeared. Because of that weakness, deeper methods became necessary. What followed was the CNN stepping in, building its own sense of patterns without manual help. Over time, it adjusted itself as network behavior shifted.

What stands out most about this project is how it brings live tracking together with visual display. Built on a FastAPI foundation, the backend uses WebSocket links

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

to keep data flowing without delay, feeding updates straight into the interface. Because information moves so quickly, users can respond sooner, making the whole setup more useful in daily operation.

What stands out is how balance matters - accuracy on one side, speed on the other. Though heavier on computing power, the CNN model finds what others miss: hidden threats, subtle patterns. That edge gives it weight in today's digital defenses.

One thing stands clear: staying flexible matters when protecting networks. When dangers online shift shape, tools that spot these changes need to grow too. What helps? Systems built on

deep learning - they adjust as situations change. Their ability to evolve makes them fit well ahead.

Putting it all together, this study highlights how shifting from classic machine learning to deep learning helps create smarter, larger-scale, yet dependable tools for securing networks. What has been introduced here sets a clear path forward for future exploration and progress in detecting intrusions.

FUTURE WORK

Although the proposed system demonstrates promising results, there are several areas where further improvements can be made to enhance its performance and applicability.

One important direction for future work is the integration of real-time packet capture. Currently, the system uses simulated network traffic, which provides a controlled environment for testing. However, implementing live packet capture using tools such as network sniffers would make the system more realistic and applicable in real-world scenarios.

Another area for improvement is the use of advanced deep learning architectures. While the CNN model performs well, combining it with models such as Long Short-Term Memory (LSTM) networks can improve the detection of temporal patterns in network traffic. Hybrid models such as CNN-LSTM can capture both spatial and sequential features, leading to better performance.

Scalability is also an important consideration for future development. As network sizes increase, the system must be able to handle large volumes of data efficiently. Deploying the system on cloud platforms and using distributed processing techniques can help improve scalability and performance.

In addition, improving the detection of zero-day attacks remains a key challenge. Future work can focus on unsupervised or semi-supervised learning techniques that can identify unknown attack patterns without relying on labeled data.

Another potential improvement is enhancing the user interface of the dashboard. Adding more visualization features, such as detailed analytics and historical trends, can provide better insights into network behavior and improve decision-making.

Finally, integrating the system with existing security frameworks and automated response mechanisms can make it more practical for real-world applications. This would allow the system not only to detect threats but also to respond to them automatically.

REFERENCES

[1] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis,''Network anomaly detection with the restricted Boltzmann machine,'' Neurocomputing, vol. 122, pp. 13–23, Dec. 2013.

[2] M. Al-kasassbeh, M. A. Abbadi, and A. M. Al-Bustanji, ''Light-GBM algorithm for malware detection,'' in Proc. Sci. Inf. Conf., 2020.

[3] S. Hajj, R. El Sibai, J. B. Abdo, J. Demerjian, A. Makhoul, and C. Guyeux,''Anomaly-based Intrusion detection systems:The requirements ,methods, measurements, and datasets,'' Trans. Emerg. Telecommun, Technol., vol. 32, no. 4, p. e4240, Apr. 2021.

[4] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, ''Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations,'' Int. J. Crit. Infrastruct. Protection, vol. 38, Sep. 2022.

[5] S. Ola-Obaado and M. A. Suleiman, ''Anomaly-based network intrusion detection using transfer learning,'' in Proc. 2nd Int. Conf. Multidisciplinary Eng. Appl. Sci. (ICMEAS), Nov. 2023.

[6] M. Mynuddin, S. U. Khan, Z. U. Chowdhury, F. Islam, M. J. Islam,M. I. Hossain, and D. M. A. Ahad, ''Automatic network intrusion detection system using machine learning and deep learning,''in IEEE MTT-S Int. Microw. Symp. Dig., Feb. 2024.