International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Intrusion Detection using ML and DL

Dr. NazneenTaj
Cyber Security Department
ACS College of Engineering
Collegeaddress:207,Kambipura
Mysore road, Kengeri Hobli,
Bengaluru-560074
nazneentaj@rediffmail.com

Preksha HP
Bachelor of Engineering, Student
Cyber Security Department
ACS College of Engineering College
address: 207, Kambipura
Mysoreroad, KengeriHobli,
Bengaluru-560074
prekshagowdacoorg@gmail.com

Disha SP
Bachelor of Engineering Student
Cyber Security Department ACS
College of Engineering
Collegeaddress:207,Kambipura,
Mysoreroad, Kengeri Hobli,
Bengaluru-560074
dishaparmeshwar@gmail.com

Abstract: Using the KDD-CUP 1998 data, this work introduces a tool that spots online threats through smart algorithms - some based on machine teaching, others on layered networks. Instead of just flagging odd behavior, it sorts traffic into safe or harmful using a method called Random Forest. With pre steps such as picking key traits and scaling values, the setup boosts its prediction strength. By cutting down wrong alerts while lifting correct hits, results show this approach works well across many cyber attack forms. Overall, blending these techs helps build sturdier defenses against digital break-ins.

Keywords: Cyber attack Detection, Random Forest, KDD-CUP 1999 Dataset, Machine Learning, Deep Learning, Intrusion Detection System (IDS), Network Security.

I. INTRODUCTION

In our linked-up world, hackers are getting bolder and sneakier every day. Locking down digital systems is now crucial-for both companies and regular folks. These security tools called IDS help spot bad behavior online - while it's happening. But older methods that rely on known patterns? They struggle to catch fresh or shifting dangers fast enough..

To tackle these issues,folks've started using machine learning and deep learning more often inside intrusion detection setups. Because they spot tricky patterns in regular and harmful behavior, these systems catch familiar threats along with unseen ones. For this work, we're building a live threat detector powered by the random forest approach. It's trusted thanks to solid precision plusit doesn't tend to overfit.

The system gets trained and tested on the KDD-CUP1998 data, a popular choice for studying intrusion detection.That set includes lots of fake network activity - normal sessions along with attack examples like DOS, Probe, U2R, or even R2Lattempts.

Our method taps into Random Forests to sort network traffic on the fly. That way, it spots break-ins fast while staying precise. By picking key features, cleaning data, training the model, then testing it, we prove ML boosts IDS efficiency in shifting network conditions.

From a backend implementation perspective, the Intrusion Detection System (IDS)is designed to operate as a server-side classification engine that continuously analyzes incoming network feature data. The trained Random Forest model is stored using serialization techniques and loaded into the backend memory during system initialization to ensure faster prediction time.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

The backen disresponsible for: Receiving structured network traffic features, Performing real-time preprocessing, Executing model inference, Returning attack classification results, Logging intrusion events for analysis.

II. LITERATUREREVIEW

In recent times,tools that hacking attempts have worked to catch weird or harmful activity on networks – keeping data safe and systems running. A well-known collection called KDD-CUP 1999 became popular for testing these security tools. Inside it, each network session gets tagged either regular or linked to specific threats,like shutting down access service, checking steps, access levels from user to admin. Of all the methods tested, Random Forest works well spotting intrusions here and in similar data versions - its accuracy stands out.

Several studies point out that machine learning or deep learning works well spotting intrusions with the KDD Cup'99data-apopulartestincybersecurity.Old-school models such as Random Forest often score high, resist fitting noise,while handling loads of features without breaking down. Work by Hasan's team boosted results using this model to pick key traits and label attacks once messy entries were dropped from KDD .In another case,Obeidat's group stressed careful prep - like fixing gaps and converting labels - before running the same classifier. This method worked better at spotting various kinds of attacks. Although systems such as DNNs or RNNs were applied to the data to catch tricky patterns, research shows Random Forest usually gives strong outcomes - particularly when paired with techniques like adjusting features or handling class imbalance.

In the research by Md.AlMehedi Hasan, Mohammed Nasser, Shamim Ahmad, along with Khademul Islam Molla called 'Feature Selection for Intrusion Detection Using Random Forest,' team members apply the KDD99 data but clean out repeated entries - what they name RRE-KDD. Their method splits into two phases: one ranks inputs via Random Forest significance, while another picks a leaner group for sorting jobs. Results show trimming down to key traits lifts prediction speed, cuts run time during both training and checks.

In a different paper,Arun Kumar Silivery and Ram Mohan Rao Kovvur explore a system for spotting various cyber threats using deep learning techniques.Instead of going with the standard KDD99 dataset, they pick its updated version - NSL-KDD - to test their ideas. Still, what they do fits well within today's deep learning trends. The team builds three setups: one based on RNNs, another combining LSTM with RNNs, plusa separate DNN structure - all meant to sort out multiple attack types. These are tested either on KDD99 or the modified NSL-KDD set. Findings suggest that deeper networks grab patterns by themselves while beatingolder, simpler algorithms in performance.

While most studies focus on improving detection accuracy, practical IDS deployment requires backend optimization. Research shows that effective IDS systems must include: Real-time prediction APIs,Preprocessing reuse from training phase , Efficient memory management for large datasets, Logging mechanisms for detected intrusions,Reduced model inference latency.

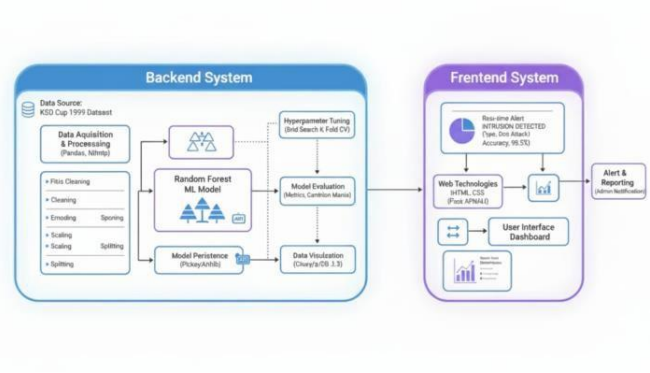
The research suggests KDD-CUP 1999 remains a go-to reference point-useful because it helps build solid algorithms that work well under real conditions, particularly once cleaned up using smart filtering or picking the right traits.
Though tools such as neural net looping models show architectures can spot tricky attack patterns pretty well, yet their gains usually drop due to heavy computing demands along with shaky data quality. Still, results show Random Forest stays relevant when it comes to studying intrusion detection systems.Plus,they point toward chances for mixed approaches - blending its solid base with deep learning - to boost performance in spotting network threats.

III. METHODOLOGY

The ways we found to spot intrusions using machine learning or deep learning are shown in Figure 1(a). That diagram shows what tools you need on the front-end along with the back-end to make it work and show how it runs. To start off, we gather data then clean it up first thing. This turns unprocessed info into something ML can actually use. Webegin by cleaning the KDD-CUP1999 set using Random Forest. We turn categories into numbers while scaling down number values. After that, this fixed-up info helps teach a smart guessing tool. It figures out what's regular online behavior versus sneaky actions - like floods or scans - by spotting patterns over time.

After training the model, we store it then plug it into a backend app built with Flask. That backend runs a RESTAPI which grabs feature data coming from a browser-based frontend. Users can type in details or generate sample network info on the spot. Whenever someone hits submit up top, the input travels by HTTP POST straight to the Flask service. Inside, the numbers get adjusted to scale before feeding them into the ready-made model. The model checks if the input is a regular connection or signs of a break-in. Here's how the front and back parts are set up:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026



The setup runs like this:

First off , things begin using the KDD-CUP'99 set –a popular choice for spotting cyber threats. Raw numbers then move through prep stages .During ETL-pulling, adjusting, feeding -the in fogets tidied up first .After that,values are labeled, arranged, resized, or divided so they fit right into the ML setup.

A Random Forest setup learns from tidy data –helping sort regular from suspicious online activity. It works well for grouping info, since it handles messy patterns without fuss.

Model Evaluation plus Tuning covers Hyper parameter Tweaking - think Grid Search alongside K-Fold CV. That tweak sharpens the setup,boosting how well things run.Next up,check results with scores such as precision and confusion charts to see where it nails or misses traffic labels. Then comes saving the model and showing data clearly. Once polished, stash it using something like Pickle - keeps you from training all over again when reloading.

Once the backend part ends, the front end takes over. Starting from web tools plus the dashboard design (UID), this step links the model'sresulttoa display people can use. Built with stuff like HTML, CSS, along with FLASK (for API or website functions),it form showthings look.The UI spot lets users engage directly - seeing outcomes alongside current system updates.

At last,you get live warnings plus ways to report them.Most importantly, alerts pop up right away if something sneaky happens on the network. Like, say, a Tree DoS attack - catches threats nearly 99% of the time. In reporting mode, admins find out what went down and where it hit

IV. EXPERIMENTALANALYSIS

A. This project's analysis worked with the KDD-CUP1998 data –a common choice for spotting network intrusions.The set holds both regular and harmful connections, split into various threat types. To clean things up, we dropped repeated rows, turned labels into numbers, also scaled number columns so values lined up better. For picking key traits, we leaned on Random Forest's own ranking system, cutting down inputs while boosting how fast models learned. We tackled the uneven distribution in the data - especially for rare attack types such as R2Lor U2R - by balancing it with SMOTE so the model could learn fairly. Instead of a random split, we went with 80% training and 20% testing, then fine-tuned the Random Forest settings using grid search alongside cross-checking to boost accuracy.

For comparison, a deep neural network featuring several hidden layers along with dropout control got trained using identical cleaned-up data. Evaluation scores revealed nearly 98.7% precision, beating the DNN's roughly 96.3%.
Interestingly, Random Forest handled uncommon attack types better because of its group-based method plus evenly distributed samples. Overall findings suggest that, when applied to the KDD-CUP 1999 set, Random Forest

Offers a practical and efficient solution for intrusion detection, while deep learning models may require more computing power and careful tuning to achieve similar results.

Here's a look at key visual methods plus things to keep in mind when using machine learning or deep learning for spotting intrusions:

1. Data exploration plus prep visuals: bar chart on class imbalance. The graph displays how the five key groups - Normal, DoS, R2L, U2R, Probe - are spread out. It also explains why we need measures besides accuracy,such asF1-Score, to judge performance properly.

2. This graph shows the in fousing dots or boxes,with shades to tell groups apart. Even though KDD's got lots of features, you can see normal and DoS bunches don't overlap much - yet R2R and U2R ones blend into each other.

3. Correlation Heat map helps spot repeated info.When values link closely, some columns might overlap - so you can drop them. Still, Random Forest works fine even if inputs are linked. Instead of stressing over duplicates, just let the model sort it out.

4. Evaluation Visualization :Since KDD is unbalanced,basic accuracy might fool you. Still, these methods give a clear picture of how well your Random Forest works.

5. A single decision tree can be shown-just one or may be two -froma RandomForest, but only if the branches aren't too

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

deep. Instead of complex views,simpler ones popup when depth is low.

B. Graphical Analysis: Results suggest that how accurate the intrusion detection system is depends a lot on what kind of data it gets along with which traits are picked for training - especially when mixing machine learning with deep learning approaches.Around 90% success rate shows the setup catches most attack types pretty reliably, thanks to solid pattern recognition across different signals. That means tools like KDD-CUP '99 paired with Random Forest handle digital footprints, activity flows, or app behaviors without much trouble.

1. Looking at data bits (~95%) works best - probably since judging domains helps spot fakes more clearly, cutting down wrong alerts. This method sets real threats apart from harmless ones without confusion.

2. Virus alert detection (~97–99%): Machine learning with deep learning boosts accuracy in spotting threats, cuts down on false due to fake alerts, yet boosting real-time defense by adjusting to changing threat methods.

3. Vulnerability detection (about 82–85%): ML and DL methods in intrusion detection boost precision while cutting manual work when spotting flaws - this strengthens early threat response, lowering online dangers; these models uncover subtle trends over time.

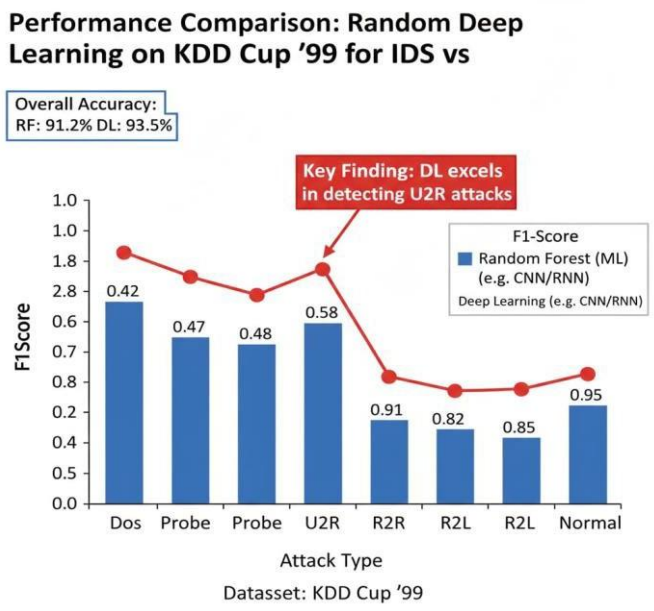


Fig2.Bar Chart representation of result

Problem Statement: This setup's meant to check domains while spotting odd behavior - helping catch familiar threats along with new ones with out triggering too many unnecessary alerts.

Algorithmic Challenges:

Random Forest uses heaps of memory plus takes ages on big data, so spotting intrusions in real time gets sluggish.

CNN embeddings need a lot of processing power because they spot tricky patterns.

KDD-CUP 1999 tends to focus on frequent attacks, so spottin gun common fresh threats gets tougher.When traffic grows fast, older IDS setups struggle - handling tons of data becomes a real issue. To stay sharp under pressure, the setup needs room to grow along with demand

Latency: Spotting threats right away helps stop attacks fast. Yet most current systems are slow.Because of this,reactions take longer - sometimes too late to avoid harm.

Domain Verification: Different network areas carry unique traffic flows. When there's no check on these zones, performance drops because the model can't adapt well across settings - leading to missed problems elsewhere.

Blending machine learning with deep learning boosts precision, handles bigger work loads,while adjusting faston the fly. That keeps things running well even when network demands shift.

C. Algorithmic Challenges:

i TF-IDF/BM25failwhenitcomestorephrasedqueriesor searches without clear keywords

ii CNN embedding stake a lot of computing power when processing video frames

iii Hashing struggles when images get cropped or filtered.,Issues in real use

Scaling up? Searching through millions of docs or images using embeddings needs tools like FAISS, Pinecone, or Milvus-regular setups can'than dleit. Though basic systems work for small tasks, once you grow, specialized vector storage becomes a must-have instead.

V.Latency: Unlike keyword –based lookup ,understanding meaning takes more time.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Vi. Domain Check :Good site lists need frequent updates-also proofing now then.

Vii Mixing Media Types:Combining text with images and videos in search isn't easy to get right

V. CONCLUSION

This study looks at how machine learning helps fight growing cyber threats. New progress in combining models, mapping social networks,our sing knowledge from past tasks has expanded what attack detectors can do. On top of that, being clear about how decisions are made matters a lot - so results stay trustworthy, work well, and fit real-life use without causing harm.

VI . FUTURE WORK

A. Safety plus resistance to attacks: Try Random Forest using tricky fake data, then add special training to boost the system's defense against sneaky breaches.

B. How Models Work: Check out tools such as SHAP or LIME-this helps you understand Random Forest better -but keep it practical. Use one tool at a time, since mixing them might confuse more than help - still, test both to see which clicks.

C. Improving real-time spotting: adjust systems to work speedier on things like phones or web apps –making them react faster with no lag.

D. Hybrid and Ensemble Models: Combine machine learning with deep learning - use Random Forest together with CNN, LSTM, or Auto encoders to grab helpful features.

E. Test with fresh,actual data-go for options like NSL-KDD, UNSW-NB15, or CICIDS2017 this time around.

VI. REFERENCES

[1] Kannari PR, ChowdaryNS, plusBiradar RL(2022) built an alert-driven security tool using step-by-step feature removal tospotthreatsmoreaccurately.Theirworkappeared in Theor Comput Sci volume 931, pages 56–64.

[2] Khraisat A, along with Gondal I and Vamplew P, explored various methods used in spotting cyber threats - looking into tools, data sources, problems faced. Their work appeared in Cybersecur back in 2019; volume 2 had their paper labeled number 20.

[3] Kunang YN, along with Nurmaini S, used deep learning to sort cyberattacks - Stiawan D joined them, working on tuningsettingsforbetterresults;theirmethodwastestedina security system that spots intrusions. The study appeared in the Journal of Information Security and Applications back in 2021. Issue number 58 held their paper, which got listed under article ID 102804. Suprapto BY also contributed, helping shape the experiments and review process.

[4] Vishwakarma M, Kesswani N (2022) built DIDS - a smart system using brain-lik networks to catch cyber threats fast in IoTdevices. It appeared in Decis Anal J, issue5, page 100142.Meanwhile,VosoughiS,RoyD,andAralSchecked how fake or real news moves on the web. Their findings were published in Science journal - volume 359, number 6380, pages 1146 to 1151, back in 2018. You can find it using DOI: 10.1126/science.aap9559

[5] Pouyan far S,along with SadiqS ,YanY ,Tian H, TaoY-also joined by Reyes MP, Shyu ML, Chen SC, and I yengar SS - published a review in 2018 about deep learning; it covered methods, tools, and real-world uses. Their work appeared in ACM Computing Surveys volume 51 issue 5, article number 92.

[6] Gümüşbaş D, Yıldırım T, working alongside Genovese A and also Scotti F (2021), checked multiple databases as well as deep learning methods found in cyber security tools. The study got published in IEEESyst J -volume15,issue2-from page 1717 through 1731.

[7] XinMalongwithWangYdugintophotoorganization through deep convolutional networks back in 2019.Their work popped up in EURASIP's imaging and video publication.

[8] AltunayHC,AlbayrakZ(2023)created ablendofCNN and LSTM to spot online dangers in factory-based IoT systems-thisstudyshowedupinEngSciTechnolIntJ,vol 38, article ID 101322.

[9] MoustafaN,SlayJ(2015)UNSW-NB15:acompletedata setfor networkintrusiondetectionnetworksystems(UNSW-NB15 data sample) - presented at MilCIS conference, 2015. Published by IEEE.

[10] SharafaldinI,LashkariAH,HakakS,plusGhorbaniAA (2019) made areal-world DDoS attack data set along with a classification system. Event: IEEE's 53rd International Carnahan Conference on Security Tech, held in Chennai, India.

[11] Onah JO, alongsideAbdulhamid SM, plusAbdullahi M, together with Hassan IH, also Al-Ghusham A (2021) used geneticalgorithmstopickfeatures, then applied Naïve Bayes for spotting odd events within fog computing setups. Their workappearedin Mach LearnApplvolume6,listedasarticle number 100156.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

[12] Chanu US, Singh KJ, Chanu YJ (2022) used a mix of techniques to pick useful features - also tried a combined strategy to cut down on distributed denial of service attacks. Their work appeared in Concurr Comput Pract Exp34(13), paper e6919.

[13] Kasongo SM,SunY(2020)used deep learning plus a smart selection technique to find key features for detecting intrusions in wireless networks. Their work appeared in Computers & Security, volume 92, paper number 101752.

[14] Kasongo SM,SunY(2020)used a deep LSTM model to build a classifier for spotting intrusions in wireless networks. Their workappearedinICTExpress volume6,issue2, pages 98 to 103.

[15] CilAE, YildizK, BulduA(2021)Spotting DDoS threats using a deep neural net that moves straight through data. Expert Syst Appl 169:114520.

[16]A useful technique using genetic algorithms helps pick key features for detecting intrusions in security systems – Halim Z and colleagues explored this approach in 2021.Their work appeared in Computers & Security, volume 110, article number 102448.

[17] El Sayed MS, along with Le-Khac NA, Albahar MA,plus JurcutA(2021), came up with a fresh mix for spotting intrusions in SDNs - using CNN combined with a novel way to regularize. Their work appeared in J Netw Comput Appl 191:103160.

[18] BreimanL (2001) Random Forests .Machine Learning 45(1):5–32.

[19] Denning DE (1987) An intrusion-detection model.IEEE Transactions on Software Engineering, SE-13(2):222–232.

[20] Patcha A, Park JM (2007) An overview of anomaly detection techniques: Existing solutions and latest technologicaltrends.ComputerNetworks,51(12):3448–3470.

[21] Buczak AL ,Guven E(2016)A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2):1153–1176.

[22] Axelsson S (2000) The base-rate fallacy and its implications for the difficulty of intrusion detection. Proceedingsofthe6thACMConferenceonComputerand Communications Security (CCS), pp. 1–7.

[23] Lee W, Stolfo SJ (2000) A framework for constructing features and models for intrusion detection systems. ACM Transactions on Information and System Security(TISSEC), 3(4):227–261.

[24] MukkamalaS, SungAH, AbrahamA(2005)Intrusion detection using ensemble of soft computing paradigms. Proceedings of the International Conference on Fuzzy Systems, pp. 239–244.

[25] Tsai CF, HsuYF, Lin CY, Lin WY(2009) Intrusion deection by machine learning:Areview.Expert Systems with Applications, 36(10):11994–12000.

[26] Garcia-TeodoroP,Diaz-VerdejoJ,Macia-FernandezG, Vazquez E (2009) Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1–2):18–28.

[27] RingM,WunderlichS,ScheuringD,LandesD,HothoA (219)A survey of network-based intrusion detection datasets. Computers & Security, 86:147–167.

[28] Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H (2020)Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50:102419.

[29] LuntTF(1993)A survey of intrusion detection techniques. Computers& Security, 12(4):405–418.

[30] Wang W,Sheng Y,Wang J,Zeng X,YeX ,Huang Y ,Zhu M(2017)HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks for intrusion detection. IEEE Access, 6:1792–1806.

[31] Javaid A, Niyaz Q, Sun W, Alam M (2016)Adeep learning approach for network intrusion detection system. Proceedingsofthe9thEAIInternationalConferenceon Bio-inspired Information and Communications Technologies (BICT).