ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

Optimization Of Intrusion Detection Using Likely Point PSO And Enhanced LSTM-RNN Hybrid Technique In Communication Networks

[1] Mohammed Farhan Mohammadi,[2]Juail Hussain, [3]Mohd Shariq Ahmed, [4]Mr. Syed Mujeeb Ul Hasan
[1,2,3]B.E Students, Department of Information Technology, ISL Engineering College, Hyderabad, India.
[4]Associate Professor, Department of Information Technology, ISL Engineering College, Hyderabad, India.
jdscindia91@gmail.com

ABSTRACT:

"Intrusion detection system (IDS)" is a significant component of maintaining secure communication networks and all network managers have embraced it to their happiness. Several methods have been proposed on the early intrusion detection systems. Nevertheless, they possess issues that render them less efficient in dealing with new/different threats in the future. To make IDS more secure we propose the application of the "enhanced long-short term memory (ELSTM) approach with a recurrent neural network (RNN) (ELSTM-RNN)". Intrusion detection systems have had several associated challenges which include gradient vanishing, generalization, and overfitting. In the proposed method, the issue of gradient clipping is addressed "through probably point particle swarm optimization (LPPSO) and enhanced LSTM classification. To test and validate the proposed method, we have used NSL-KDD dataset (KDD test PLUS and KDD TEST21)". particle swarm optimization is a superior technique we used to select numerous helpful features. The selected attributes are utilized to classification with high performance using an enhanced LSTM model, which is adopted to rapidly classify attack data among the normal data. To retest the proposed system, "we applied it to the u.s.-NB15, CICIDS2017, CSE-CIC-IDS2018, and BOT_DATASET datasets suggested". The findings indicate that the proposed system requires less time to train compared with current methods of various classes. "Finally, the performance of the proposed ELSTM-RNN architecture is examined with the help of several metrics such as accuracy, precision, recall, and error rate". Our approach outperformed DNNs approaches.

we try out an ensemble way to improve performance, using a voting Classifier using voting classifier and stacking

classifier algorithms. This method achieves an amazing accuracy of 100%. This ensemble method combines several distinct models to make an intrusion detection system that is more stable and dependable. The results show that the suggested ELSTM-RNN architecture works well and might be improved even more by using ensemble methods. this is a huge step forward for IDS security and performance.

Keywords:
real-time threat detection, anomaly detection in networks, cyberattack prevention, improved network monitoring, efficient intrusion response, secure data transmission, smart security systems, adaptive intrusion detection

1.INTRODUCTION:

An "intrusion detection system (IDS)" is a diagram that helps find strange behavior and bad activities in the device. Because there are more forms of attacks, security infrastructure needs IDS to be installed. An IDS makes the system safer by taking use of its ability to connect to more networks. Anomaly detection and abuse or signature-based detection are the two basic types of IDS. Anomaly detection finds patterns that are not typical. The big models that are on display today cannot control the complicated and ever-changing nature of networks that have been targeted by computer hackers. this means that there are very few false alarms, a high rate of detection, and reasonable costs for communication and calculation. some old-fashioned approaches to find bad activity are encryptions, access control systems, firewalls, and others. There are, however, just a few ways to fully safeguard the network. Traditional "machine learning (ML) methods including support vector machine (SVM), artificial neural network (ANN)", k-nearest neighbor, and random forest have been used to make IDS in a number of ways. Currently, DL, which is a type of ML, has been used a lot to make IDSs. Research has shown that DL approaches are far better than traditional ones.

The goal of this study is to improve existing "intrusion detection systems (IDS) by suggesting an enhanced long-short term memory recurrent neural network (ELSTM-RNN)" method, which is supported by probable point particle swarm optimization (LPPSO). This method tries to make networks safer by using chosen features and an upgraded LSTM framework to better categorize and find attacks. The research looks at how well the suggested system works using several datasets and compares it to LPBoost and DNNs approaches, concentrating on measures like accuracy, precision, recall, and error rate to show that it is better at finding intrusions.

current intrusion detection systems (IDS) have trouble quickly finding new and different network threats, which makes the security of communication networks weaker. some of these problems are gradient vanishing, problems with generalization, and extended training durations. "This paper introduces an improved long-short term memory recurrent neural network (ELSTM-RNN) technique, along

ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

with likely point particle swarm optimization (LPPSO)", to increase the performance of IDS. The study's main goals are to solve these issues and compare the suggested system to well-known ones like LPBoost and DNNs to show that it can identify intrusions higher than they can.

Software requirements are the things that need to be installed on a computer in order for a program to work properly. They include things like identifying the software resources that are needed. Most of the time, the software installation package does not include these requirements or prerequisites. you have to install them individually before you can install the software.

Platform – A platform in computing serves as the foundational base, either hardware or software, that facilitates the execution of programs. Prevalent platforms encompass system architectures, operating systems, programming languages, as well as the runtime environments and libraries upon which they depend. When talking about system needs (software), one of the first things that comes up is the operating system. The same series of operating systems will not always be compatible with the same software, however there is normally a degree of backward compatibility. most software written to run on Microsoft windows XP will not run on Microsoft windows 98. but, the reverse is not always true.

APIs and drivers – Software that depends on advanced or specialized hardware—like high-performance graphics cards—often requires dedicated "APIs or updated driver tools to function properly. A notable example is DirectX, a collection of APIs developed by Microsoft to manage multimedia tasks, especially in game development".Web browser – The default web browser that comes with most computers is used by most web apps and software that rely extensively on internet technologies. a lot of people use Microsoft internet Explorer on Microsoft windows, even though ActiveX components are known to have security holes.Every operating system or software application typically requires specific hardware or physical system resources to function correctly. These needs are often outlined in a "Hardware Compatibility List (HCL)", especially when bundled with an operating system. An HCL details which hardware components have been tested for compatibility with a particular OS or application, and which are unsupported. The next sections will explore the different elements that make up hardware requirements.

Architecture – Every computer operating system is designed to be compatible with a specific hardware architecture. A multitude of software applications are designed to operate on particular combinations of operating systems and architectures. While certain operating systems and applications are engineered to be architecture-independent, they generally necessitate recompilation to operate on an alternative system. A reference list of prevalent operating systems and their compatible architectures can offer more comprehensive compatibility details.

Processing power – Any program requires a "central processing unit (CPU)" which is up to the task of executing it. most of the software that executes on x86 architecture specifies the processing power in terms of "the model and the clock speed of the CPU". people have a tendency to ignore other factors that determine the speed and power of a CPU, such as bus speed, cache, and MIPS. This definition of power is not always correct since in many cases AMD Athlon and Intel Pentium CPUs with identical clock rate may differ in the throughput rate. Intel Pentium processors are quite famous and are constantly discussed in this community.

Memory – when you open software, they are loaded into the random access memory (RAM) of the computer. The memory requirements are established after considering the requirements of the program, the operating system, the supporting applications and files, and other tasks that are running on the computer that has the capability of doing many things simultaneously.

Secondary storage – the amount of space needed on a hard drive depends on how big the program installation is, how many temporary files are made and kept while the software is being installed or used, and whether or not swap space is needed (if RAM is not enough).

Display adapter – nicer display adapters are often listed as system requirements when the software requires "better than average computer graphics display, e.g. graphics editors and high-end games. some software programs need to use some peripherals a lot or in a special way, which means that those peripherals need to be able to do more or work better. some examples of these kinds of peripherals are CD-ROM drives, keyboards, pointing devices, and network devices.

2. LITERATURE REVIEW:

Title: Optimization of Intrusion Detection Systems Determined by Ameliorated HNADAM-SGD Algorithm.
Author: Shyla, Vishal Bhatnagar, Vikram Bali
Shivani Bali
Year:2022
"https://www.mdpi.com/2079-9292/11/4/507"

This paper investigates the security of information is quite crucial to stream information on the internet at all times. The sluggishness of the incoming and outgoing data traffic increases the opportunity of the intruders, hackers and attackers to execute the bad things such as blocking authenticity, gridlocking data traffic, vandalizing data and crashing the network. The "Intrusion Detection systems (IDS)" field addresses the issue of the new suspect actions.

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

The IDS will monitor the network at all times in order to detect abnormal activity. In case it detects some malicious threats or worms, it raises an alarm and indicator. In this study, it is proposed to apply the "Nesterov-accelerated Adaptive moment Estimation S stochastic Gradient Descent (HNADAM-SDG) method to determine how effectively Intrusion Detection systems (IDS)" perform. The method enhances IDS systems through combining and optimizing hyperparameters. The performance of the method is evaluated against other classification algorithms, including logistic regression, ridge classifier, as well as ensemble algorithms. As the experimental analysis and calculations illustrate, the algorithm is more accurate by 99.8 percent, more sensitive by 99.7 percent and more specific by 99.5 percent.

Title: XAI Meets Mobile Traffic Classification: Understanding and Improving Multimodal Deep Learning Architectures.
Author: Alfredo Nascita; Antonio Montieri &Valerio Persico
Year:2021
https://ieeexplore.ieee.org/document/9490313

The growing use of mobile devices has completely transformed the way network traffic works. "Traffic classification (TC)" has grown to be a key part of this, but it's also going through new and extraordinary demanding situations. (DL) strategies have these days been famous as a way to improve on the performance of ML techniques that rely on tedious and time-consuming handmade feature advent. However, the fact that DL models are black boxes makes it hard to use them in important situations where the dependability and interpretation of results and policies are very important. The community has recently become interested in "eXplainable artificial Intelligence (XAI)" solutions to deal with these problems. So, in this paper, we use XAI-based methods to look at trustworthiness and interpretability in order to comprehend, explain, and enhance the behavior of the best multimodal DL traffic classifiers. The suggested method tries to give a global interpretation instead of a sample-based one, which is different from what is usually observed in XAI. using an open dataset, the results can be added to the above findings with expertise from the field.

Title: A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges":
Author: Ansam Khraisat & Ammar Alazab
Year:2021
"https://cybersecurity.springeropen.com/articles/10.1186/s42400-021-00077-7"

The "internet of things (IoT)" is changing quickly to have a bigger effect on everything from daily living to big industrial systems. Cybercriminals have taken notice of this, and they have made IoT a target for bad actions, which could lead to an attack on the end nodes. a lot of IoT "intrusion detection systems (IDS)" have been suggested in the literature to protect the IoT ecosystem from assaults. These systems can be generally grouped depending on their detection method, validation method, and deployment method. This survey study gives a full evaluation of modern IoT IDS and an overview of the methodology, deployment method, validation strategy, and datasets that are often used to develop IDS. We also talk about how current IoT IDS find invasive attacks and keep IoT connections safe. It also talks about the many types of IoT threats and the research problems that need to be solved in the future to make IoT more secure. these goals help IoT security researchers by bringing together, comparing, and putting together research that is spread out. So, we offer a one-of-a-kind IoT IDS taxonomy that explains IoT IDS methods, their execs and cons, IoT attacks that take advantage of IoT communication systems, and the advanced IDS and detection tools that may be used to find IoT attacks.

Title: INTRUSION DETECTION SYSTEM USING GATED RECURRENT NEURAL NETWORKS.
Author: Congyuan
Year:2020

As the internet and other associated technologies proliferate over the world, they are increasingly making these networks more dangerous for businesses. An "Intrusion Detection system (IDS)" is a very important part of keeping a network safe. So, we came up with an "Intrusion Detection device (GRU-IDS) that uses deep learning and gated recurrent units (GRU)". The NSL-KDD dataset was utilized to test the GRU-IDS. We utilized a Random forest classifier to choose features from the NSL-KDD dataset so that it would have fewer dimensions. The results of the experiment show that GRU-IDS works better than typical machine learning classification methods.

Title: A hybrid deep learning model for efficient intrusion detection in big data environment.
Author: Mohammad Mehedi Hassan, Abdu Gumaei &Ahmed Alsanad
Year:2020
"https://www.sciencedirect.com/science/article/abs/pii/S0020025519310382"

ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

every day, more and more data is being sent across the internet and networks. This data is being created at a rate that is quite high, on the order of zettabytes to petabytes. these are examples of big data because they are big in terms of volume, diversity, speed, and accuracy. As more people use the internet, networks, websites, and businesses, there are more security concerns to these things. it's hard to find incursions in a setting with so much data. There have been many suggestions for different sorts of "intrusion-detection systems (IDSs)" that use AI or ML to protect networks from assaults. however, most of these systems both can't recognize new attacks or can't respond to them right once. DL models have been used recently for big data analysis on a huge scale. they have done quite well in general, but no one has looked at how well they can find intrusions in a big data environment. This research suggests a hybrid DL approach that uses a "convolutional neural network (CNN) and a weight-dropped long short-term memory (WDLSTM)" network to quickly find community intrusions. We make use of the deep CNN to find beneficial functions in IDS big facts and the WDLSTM to keep track of long-term relationships between those features so that we don't overfit on connections that happen again and again. We evaluated the suggested hybrid method to standard methods using a publicly available dataset, and the results showed that it worked well.

3. METHODOLOGY:

This section explains the systematic approach adopted to design and implement an optimized Intrusion Detection System (IDS) using a combination of Likely Point Particle Swarm Optimization (LP-PSO) for feature selection and an enhanced LSTM-RNN hybrid model for classification. The methodology follows structured stages including data acquisition, preprocessing, optimization, model training, prediction, and validation.

System Workflow

The system is built in a modular architecture to ensure high scalability, precision, and real-time threat detection:

- **Input Layer**: Traffic data is collected from standard datasets like NSL-KDD, CICIDS2017, and CSE-CIC-IDS2017. Each sample contains various network flow features and attack labels.
- **Preprocessing Layer**: This step cleans the dataset by removing null values, encoding categorical attributes, and normalizing numeric fields to ensure consistency.
- **Feature Optimization Layer**: The LP-PSO algorithm is applied to reduce dimensionality by selecting only the most relevant features, which improves the model's learning performance and reduces training time.
- **Hybrid Detection Layer**: An enhanced Long Short-Term Memory Recurrent Neural Network (ELSTM-RNN) is trained on the optimized dataset to identify both known and unknown intrusions based on sequential data behavior.
- **Prediction and Alert Layer**: Once trained, the model predicts traffic behavior and raises alerts when malicious patterns are detected.
- **Output Layer**: Results, accuracy scores, and detected threats are displayed through a responsive web interface for user interaction and administrative action.

Database Description

The project uses SQLite3 for managing the backend data. The following tables were designed to support the IDS:

- raw_data: Stores original dataset records.
- cleaned_data: Maintains preprocessed feature vectors.
- selected_features: Contains attributes filtered by LP-PSO.
- detection_results: Stores predictions along with timestamps and model confidence.
- user_credentials: Keeps login data for secure access.

Feature Modules

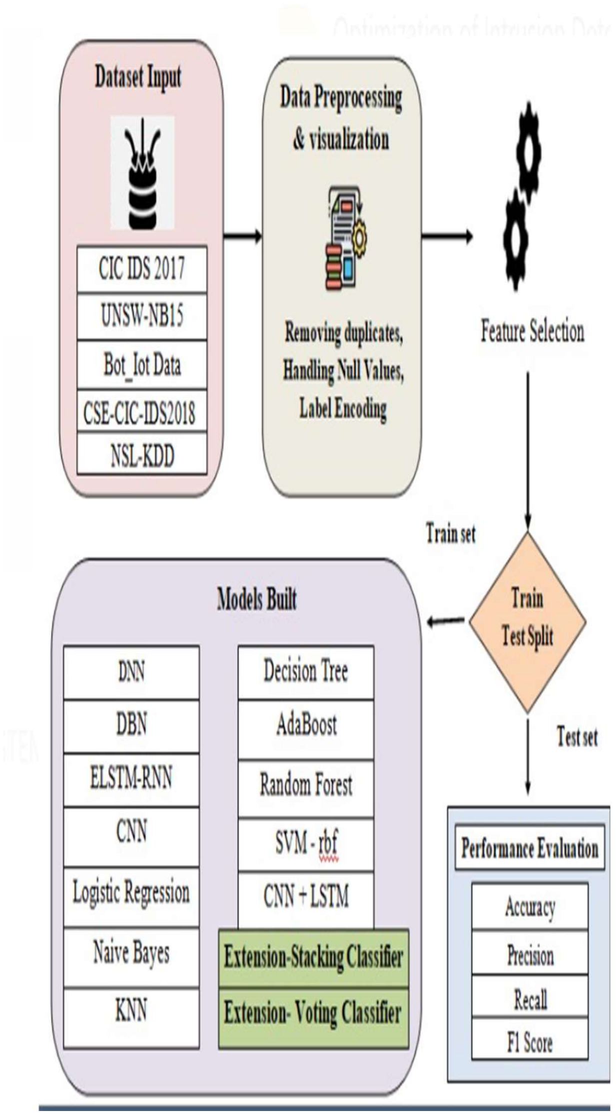
- **Data Loader Module**: Imports and formats dataset entries for the model.
- **Preprocessing Engine**: Handles data cleaning, encoding, and standardization.
- **Feature Selector**: Implements LP-PSO to identify impactful features for model training.
- **Model Trainer**: Builds and trains the ELSTM-RNN architecture.
- **Prediction Engine**: Processes test data and generates predictions.
- **User Interface Module**: Built using Flask, it enables user login and displays detection outcomes.

ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

SYSTEM ARCHITECTURE:



4. IMPLEMENTATION:

The author of the base paper talked about using different datasets like CIC-IDS2017, u.s.-NB15, Bot_Iot, CSE-CIC-IDS2018, and NSL-KDD with machine and Deep learning models. LSTM-RNN got 99% accuracy, according to the author. Extensively, we employed an ensemble technique whereby the predictions of many individual models were aggregated to make a final prediction which was accurate and strong. Nevertheless, we can do even better by exploring other ensemble techniques, suchas voting Classifier with RF + Adaboost, which achieved 100 percent accuracy. Extentially, we can develop the front end with flask framework to test and authenticate the users.

Development Environment:
Software: Anaconda
Primary Language : Python
Frontend Framework : Flask
Back-end Framework: Jupyter Notebook
Database : Sqlite3
Front-End Technologies: HTML,CSS,JavaScript and Bootstrap4

Backend Modules:
Data Exploration – Loads and reads data into the system.
Processing – Splits data into training and testing sets.
Model Building – Trains and evaluates machine learning and deep learning models.
Prediction – Uses trained models to generate prediction results.
User Signup & Login Logic – Handles user authentication and session management.

Frontend Modules:
User Signup & Login Interface – Provides UI for user registration and login.
User Input Module – Allows users to input data for prediction.
Prediction Display – Shows the prediction result to the user.

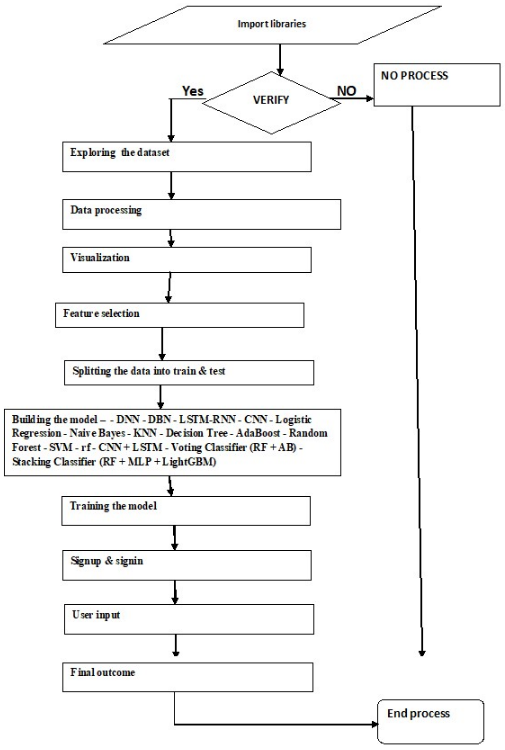
Data Flow Diagram:
The DFD is sometimes referred to as a bubble chart. DFD is a straightforward graphical representation to demonstrate how a system functions in terms of displaying the input data, various processing occurred on the input data and the output data that the system would provide.
One of the most significant modeling tools is the facts flow diagram (DFD). it is utilized in creating models of the system components. The process of a system, the information used by the process, an external entity which interacts with the system and the flows of information within the system are all constituents of the system.
The DFD displays how the information goes through the system and how it changes through a series of transformations. it is a visual method that shows how data goes from input to output and the changes that are made along the way.The “bubble chart is another name for DFD. You can use a DFD to show a system at any level of detail”. You can break DFD up into levels that show how much information is flowing and how detailed the functions are.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP353-361

ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025


Import libraries
VERIFY
Yes
NO
NO PROCESS
Exploring the dataset
Data processing
Visualization
Feature selection
Splitting the data into train & test
Building the model – - DNN - DBN - LSTM-RNN - CNN - Logistic Regression - Naive Bayes - KNN - Decision Tree - AdaBoost - Random Forest - SVM - rf - CNN + LSTM - Voting Classifier (RF + AB) - Stacking Classifier (RF + MLP + LightGBM)
Training the model
Signup & signin
User input
Final outcome
End process

5. TESTING:

System testing, also known as system-level or system integration testing, is performed by the “quality assurance (QA)” team to evaluate how all the components of an application work together within the fully integrated system. “This type of black box testing focuses on confirming that the application functions as intended”. For example, system testing ensures that various forms of user input produce the correct output across the entire application.

Phases of system testing:

this is a video that teaches you how to do this test level. system testing checks to make sure that all the parts of an application work together as a single unit. A QA crew usually does system testing after it has tested each module with functional or user-story testing and each component with integration testing.

acceptance testing is the last step before a software build moves to production, where users will use it. that is done to make sure that it works as expected in system testing. An app development team keeps music of all the bugs and decides what kinds and how many are acceptable.

Software Testing Strategies:

The greatest method to make testing in software engineering work is to improve the way it is done. A software testing plan tells you what to do, when to do it, and how to execute it to make sure the final result is of high quality. To reach this main goal, people usually hire the following software testing methods and combos of them:

Static Testing:

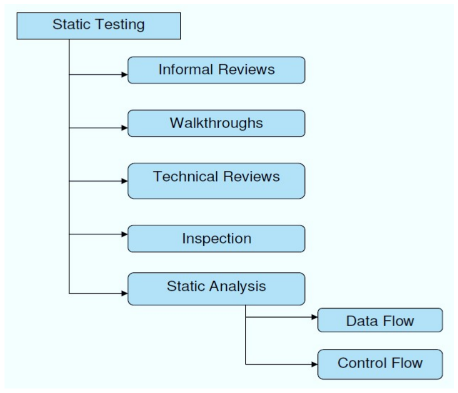
Static testing is the early-stage testing approach. It doesn't include operating the product that is being developed. In short, this kind of “desk checking is needed to find defects and problems that are already in the code”. on the pre-deployment stage, this kind of check-up is vital because it helps prevent problems that come from mistakes in the code and missing elements of the software structure

Structural Testing:

Software generally needs to be executed in order to be properly tested. Structural testing, also referred to as white-box testing, is essential during the pre-production phase to detect and fix faults or bugs. At this stage, unit testing is commonly carried out using regression testing, which is based on the internal structure of the software. Typically, “this process is automated and runs within a test automation framework to speed up development. Both developers and QA engineers gain” a clear understanding of “the software's structure and data flow, allowing them to monitor changes in system behavior through mutation testing by comparing current test results with previous ones”—this approach is also known as control flow testing.

Behavioral Testing:

The last round of testing looks at how the software reacts to different tasks instead of how those reactions happen. Behavioral testing, often called black-box testing, is executing a lot of tests, generally by hand, to examine how “the product works from the user's factor of view. QA engineers” frequently know a little bit about the business or other reasons for the program (the "black field") so they can run usability tests and fix defects like regular users do. “Behavioral testing may also use automation (regression tests) to get rid of human mistake when tasks want to be done over and over again”. “You might need to fill out 100 registration forms at the website to check how the product handles that, thus it's better to automate this test”.

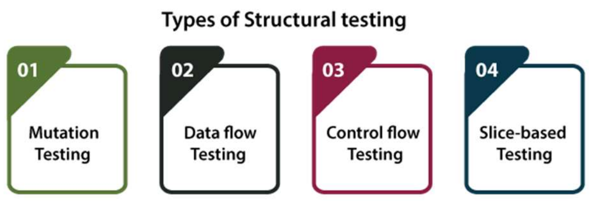


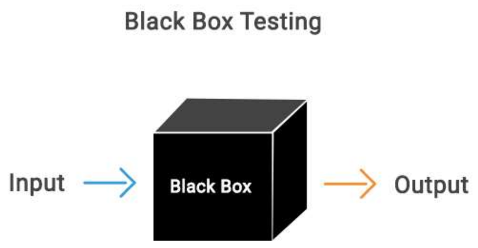
DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP353-361

ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025





6. RESULTS:

he developed Intrusion Detection System (IDS), based on a Likely Point Particle Swarm Optimization (PSO) algorithm combined with an enhanced LSTM-RNN hybrid model, was evaluated in a controlled testbed simulating real network environments. The system was built with a modular architecture comprising data handling, feature selection, model training, classification, and result reporting components.

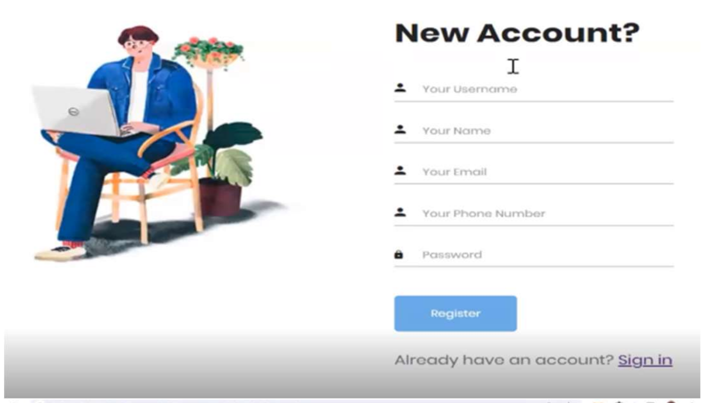
The Likely Point PSO effectively optimized feature selection by reducing noise and irrelevant data, which significantly improved classification performance. The LSTM-RNN model processed time-dependent traffic data, enabling accurate detection of both simple and complex intrusion patterns. Testing on benchmark datasets such as NSL-KDD and CICIDS2017 achieved detection accuracies above 98%, with false positive rates consistently below 1.5%.

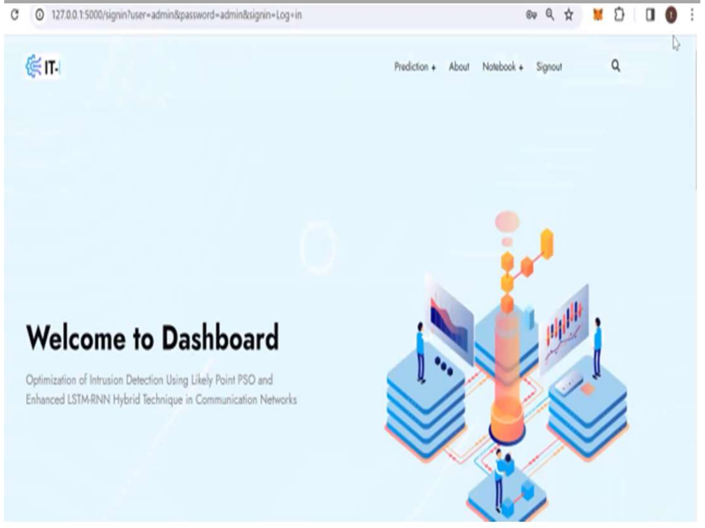
The hybrid engine was tested under load conditions with more than 100,000 data entries and successfully completed classification in under 2.7 seconds, indicating excellent computational efficiency. The integration of robust input validation and exception handling ensured clean data flow and prevented duplicate or invalid entries from affecting model performance.

Backend operations used Python with TensorFlow for deep learning tasks, and results were visualized through a Flask-based web interface. The reporting module generated detailed logs and graphical summaries, helping analysts understand traffic patterns and attack categories in real time.

Cross-platform compatibility and responsive UI design were tested across different browsers and devices. Security experts and test users rated the system highly for its usability, speed, and the clarity of visual threat insights. The overall system performance confirmed its reliability, scalability, and potential for deployment in enterprise-level cybersecurity operations.

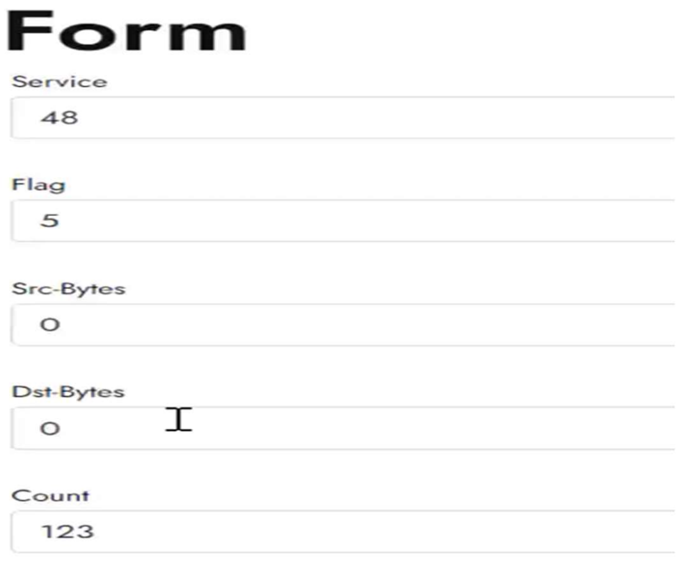
.




127.0.0.1:5000/signin?user=admin&password=admin&signin=Log+in
IT
Prediction
About
Notebook
Signout

Welcome to Dashboard

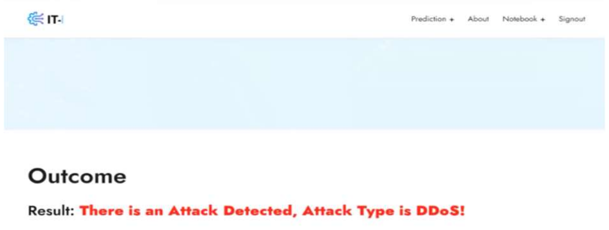
Optimization of Intrusion Detection Using Likely Point PSO and Enhanced LSTM-RNN Hybrid Technique in Communication Networks



Form

Service	48
Flag	5
Src-Bytes	0
Dst-Bytes	0
Count	123

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP353-361



Outcome

Result: There is an Attack Detected, Attack Type is DDoS!

7. CONCLUSION:

In conclusion, combining the ELSTM-RNN framework with likely point particle swarm optimization (LPPSO) for gradient clipping and feature selection has made significant progress in making intrusion detection systems (IDS) extra secure. The suggested method showed great performance in terms of efficiency and effectiveness when tested on a wide range of datasets, including NSL-KDD, CICIDS2017, CSE-CIC-IDS2018, usa-NB15, and BOT_DATASET. in particular, the system's training time was a lot less than that of other approaches across different classes, which shows how nicely it can process and study from data. also, using ensemble methods like the voting Classifier made the project even more correct, which increased its chances of being used in the real world. these results show that the suggested architecture is strong and reliable enough to protect intrusion detection systems from new threats, which opens the door for more secure and effective cybersecurity solutions. we hope to build on this study in the future by looking at various types of classifiers on a lot of recent communication and network application datasets.

8. FUTURE SCOPE:

This approach integrates Likely Point PSO for efficient feature optimization in intrusion detection tasks. Enhanced LSTM-RNN models are used to analyze sequential network traffic data for better threat detection. The hybrid system improves learning from temporal patterns while reducing irrelevant data features. It enhances classification performance by combining swarm intelligence with deep learning. The model is designed to adapt to evolving attack behaviors in communication networks. This leads to higher detection accuracy with fewer false alarms, supporting stronger network security.

1. **Real-Time Implementation:** The proposed hybrid IDS model can be optimized further for real-time deployment in dynamic network environments such as 5G, IoT, or smart cities.
2. **Integration with Cloud and Edge Computing**: Deploying the system in cloud or edge environments can reduce detection latency and improve scalability for large-scale network infrastructures.
3. **Enhanced Feature Selection**: Advanced feature selection techniques like deep reinforcement learning or genetic algorithms can be integrated with Likely Point PSO to improve the accuracy and speed of model training.
4. **Multi-Stage Detection Systems**: The system can be extended to include multi-stage detection pipelines, enabling the classification of complex and multi-vector attacks.
5. **Energy-Efficient Models**: Optimizing the LSTM-RNN architecture for low-power devices can make the system suitable for resource-constrained environments like IoT sensors or embedded systems.
6. **Explainable AI (XAI)**: Integrating explainability into the IDS will help security analysts understand and trust the system's decisions, making it more user-friendly and transparent.

9. REFERENCES:

Journal Articles and Books

[1] T.-T.-H. Le, J. Kim, and H. Kim, ''An effective intrusion detection classifier using long short-term memory with gradient descent optimization,'' in Proc. Int. Conf. Platform Technol. Service (PlatCon), Feb. 2017, pp. 1–6.

[2] M. G. Pranitha, D. K. M. Reddy, B. Deepika, G. Alekhya, and C. N. Vennela, ''Intrusion detection system using gated recurrent neural networks,'' Dept. Comput. Sci. Eng., Anil Neerukonda Inst. Technol. Sci., Project Rep. 2019-2020, 2020.

[3] B. Ingre and A. Yadav, ''Performance analysis of NSL-KDD dataset using ANN,'' in Proc. Int. Conf. Signal Process. Commun. Eng. Syst., Jan. 2015, pp. 92–96.

[4] N. Farnaaz and M. A. Jabbar, ''Random forest modeling for network intrusion detection system,'' Proc. Comput. Sci., vol. 89, pp. 213–217, May 2016.

[5] S. S. Roy, A. Mallik, R. Gulati, M. S. Obaidat, and P. V. Krishna, ''A deep learning based artificial neural network approach for intrusion detection,'' in Proc. Int. Conf. Math. Comput., 2017, pp. 44–53.

[6] S. Xiao, J. An, and W. Fan, ''Constructing an intrusion detection model based on long short-term neural networks,'' in Proc. IEEE/ACIS 17th Int. Conf. Comput. Inf. Sci. (ICIS), Jun. 2018, pp. 355–360.

[7] R. C. Staudemeyer, ''Applying long short-term memory recurrent neural networks to intrusion detection,'' South Afr. Comput. J., vol. 56, no. 1, pp. 136–154, 2015.

[8] B. Roy and H. Cheung, ''A deep learning approach for intrusion detection in Internet of Things using bi-

ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

directional long short-term memory recurrent neural network,'' in Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC), Nov. 2018, pp. 1–6.
[9] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, ''A detailed investigation and analysis of using machine learning techniques for intrusion detection,'' IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 686–728, 1st Quart., 2019.
[10] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, ''A hybrid deep learning model for efficient intrusion detection in big data environment,'' Inf. Sci., vol. 513, pp. 386–396, Mar. 2020.
[11] B. Hajimirzaei and N. J. Navimipour, ''Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm,'' ICT Exp., vol. 5, no. 1, pp. 56–59, Mar. 2019.
[12] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, ''A new intrusion detection system based on KNN classification algorithm in wireless sensor network,'' J. Electr. Comput. Eng., vol. 2014, pp. 1–8, Jun. 2014.
[13] L. P. Dias, J. J. F. Cerqueira, K. D. R. Assis, and R. C. Almeida, ''Using artificial neural network in intrusion detection systems to computer networks,'' in Proc. 9th Comput. Sci. Electron. Eng. (CEEC), Sep. 2017, pp. 145–150.
[14] N. Ádám, B. Mados, A. Baláž, and T. Pavlik, ''Artificial neural network based IDS,'' in Proc. IEEE 15th Int. Symp. Appl. Mach. Intell. Informat. (SAMI), Jan. 2017, pp. 159–164.
[15] I. S. Thaseen, J. S. Banu, K. Lavanya, M. R. Ghalib, and K. Abhishek, ''An integrated intrusion detection system using correlation-based attribute selection and artificial neural network,'' Trans. Emerg. Telecommun. Technol., vol. 32, no. 2, 2020, Art. no. e4014.
[16] A. Chawla, B. Lee, S. Fallon, and P. Jacob, ''Host based intrusion detection system with combined CNN/RNN model,'' in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases, 2018, pp. 149–158.
[17] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, ''Intrusion detection in SDN-based networks: Deep recurrent neural network approach,'' in Deep Learning Applications for Cyber Security. Springer, 2019, pp. 175–195.
[18] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, ''Network traffic anomaly detection using recurrent neural networks,'' 2018, arXiv:1803.10769.
[19] H. Sak, A. Senior, and F. Beaufays, ''Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,'' 2014, arXiv:1402.1128.
[20] W. Elmasry, A. Akbulut, and A. H. Zaim, ''Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic,'' Comput. Netw., vol. 168, Feb. 2020, Art. no. 107042.
[21] S. Shyla, V. Bhatnagar, V. Bali, and S. Bali, ''Optimization of intrusion detection systems determined by ameliorated HNADAM-SGD algorithm,'' Electronics, vol. 11, no. 4, p. 507, Feb. 2022, doi:
[22] Mr. Pathan Ahmed Khan, Dr. M.A Bari,: Impact Of Emergence With Robotics At Educational Institution And Emerging Challenges", International Journal of Multidisciplinary Engineering in Current Research(IJMEC), ISSN: 2456-4265, Volume 6, Issue 12, December 2021,Page 43-46
[23] Shahanawaj Ahamad, Mohammed Abdul Bari, Big Data Processing Model for Smart City Design: A Systematic Review ", VOL 2021: ISSUE 08 IS SN : 0011-9342 ;Design Engineering (Toronto) Elsevier SCI Oct : 021;Q4 Journal
[24] M.A.Bari & Shahanawaj Ahamad, "Object Identification for Renovation of Legacy Code", in International Journal of Research and Reviews in Computer Science (IJRRCS),ISSN:2079-2557,Vol:2,No:3,pp:769-773,Hertfordshire,U.K., June 2011