ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Stacked Autoencoder-Based Intrusion Detection System to Combat Financial Fraudulent

N. MOUNIKA, Assistant Professor, Department of MCA, Chirala Engineering College, Chirala, Nakkalamounika2121@gmail.Com

BANDARU LAKSHMI SRINIVAS, PG Student – MCA, Dept of MCA, Chirala Engineering College, Chirala, B.Laxmisrinivas2000@gmail.Com

ABSTRACT: This paper addresses the burgeoning cybersecurity risks in IoT environments and digital financial transactions by proposing a robust intrusion detection system (IDS). Leveraging advanced deep learning techniques, the IDS employs a stacked autoencoder (AE) and deep neural network (DNN) architecture. The stacked AE autonomously learns crucial features from input network records in an unsupervised manner, enhancing subsequent classification tasks' efficiency. Subsequently, the supervised DNN extracts deep-learned features for accurate intrusion classification. Evaluation on KDDCup99 and NSL-KDD datasets demonstrates exceptional performance, with the CNN + LSTM ensemble achieving an impressive 99.9% accuracy. Further enhancements involve exploring additional ensemble techniques like CNN and CNN + LSTM models, yielding promising accuracy improvements. Additionally, a user-friendly front-end interface developed using the Flask framework enables seamless user interaction and authentication. This research contributes to bolstering cybersecurity in IoT environments and digital transactions, demonstrating the efficacy of deep learning and ensemble techniques in intrusion detection. The system's practical applicability is underscored by its user-friendly interface, paving the way for adoption in real-world scenarios. Future endeavors aim to refine ensemble strategies and expand system capabilities for broader deployment scenarios.

Index Terms—Deep neural network (DNN), digital financial service, Internet of Things (IoT), intrusion detection system (IDS), stacked autoencoder (AE).

1. INTRODUCTION:

The pervasive integration of the Internet into our daily lives has brought about a significant transformation in how we interact, learn, and work. However, this deep entrenchment of the Internet also introduces a host of security concerns, particularly evident in the realm of financial transactions—a cornerstone of modern life, especially within the context of smart digital environments that support urbanization and industrialization [1].

In these smart digital environments, characterized by the proliferation of various Internet of Things (IoT) devices, the landscape of financial services is rapidly evolving [1]. As the number of IoT services continues to increase, so does the complexity of interconnected networks. While this

interconnectedness facilitates seamless communication and data exchange, it also exposes vulnerabilities that can be exploited by malicious actors [1].

The reliance on wired or wireless networks in smart digital financial services exposes these systems to a plethora of security threats. An intruder, leveraging the abundance of data traffic, can easily masquerade as a legitimate service provider, posing a significant challenge for the detection of network attacks, particularly those that are unexpected or novel [1].

The detection and mitigation of such network intrusions constitute a critical technological problem, one that demands innovative solutions. Intrusion Detection Systems (IDS) play a pivotal role in identifying and responding to these threats, whether they are ongoing attacks or have already occurred [1]. These attacks, ranging from human-made to machine-generated, are becoming increasingly sophisticated, exploiting vulnerabilities in hardware, applications, and network topologies, including advancements in IoT technologies [1].

Citing examples such as the Yahoo data breach, which resulted in substantial financial losses, and the Bitcoin hack, which underscored the dire consequences of cyber-threats, it is evident that the stakes are high [2]. Moreover, the continuous emergence of advanced algorithms further complicates the cybersecurity landscape, necessitating robust and adaptive defense mechanisms [1].

In the realm of IDS, two primary systems emerge: Network Intrusion Detection (NID) and Host-Based Intrusion Detection (HID), each targeting disruptive behaviors [3]. NID systems monitor network traffic, utilizing network devices to analyzebehavior and detect potential risks, while HID systems operate locally, scanning diverse log files for signs of intrusion [3].

Traditionally, IDS relied on a combination of signature-based detection and anomaly-based detection to identify threats. Signature-based detection involved predefined patterns and filters, while anomaly-based detection utilized heuristic tools to identify deviations from normal behavior[3]. While effective against known threats, these approaches often fell short in detecting novel or unexpected attacks, resulting in high false-positive rates.

To address these shortcomings, the field of IDS has turned to deep learning, a subset of machine learning, which has demonstrated remarkable capabilities in learning intricate patterns and features from vast datasets [4]. However, existing deep learning-based IDS face challenges such as high false-positive rates for unknown attacks, lack of transferability across datasets, and inadequate documentation [5].

In light of these challenges, this paper presents a novel approach to intrusion detection leveraging deep learning techniques, with a focus on stacked autoencoders (AE) and deep neural networks (DNN) [1]. By harnessing the unsupervised learning capabilities of stacked AEs to extract essential features from network records and the supervised learning in DNNs for accurate classification, the proposed IDS aims to enhance detection capabilities while minimizing false positives.

Furthermore, this study emphasizes the importance of evaluation on common open datasets and documentation of model performance to address the limitations of existing deep learning-based IDS [1]. By addressing these gaps, the proposed IDS seeks to offer a robust and effective solution to the cybersecurity challenges posed by the evolving landscape of IoT and digital financial transactions.

In the subsequent sections, we delve deeper into the methodology, experimental evaluation, and results of the proposed IDS, followed by discussions on the implications of our findings and avenues for future research.

2. LITERATURE SURVEY

In recent years, the escalating threat landscape in cyberspace has underscored the critical importance of developing robust and intelligent Intrusion Detection Systems (IDS) to safeguard networks from unauthorized access and malicious activities. A plethora of research endeavors have been dedicated to exploring various techniques, ranging from traditional machine learning to cutting-edge deep learning, aimed at enhancing the capabilities of IDS. This literature survey aims to provide a comprehensive overview of some seminal studies in this domain, shedding light on their methodologies, findings, and contributions.

Vinayakumar et al. [3] proposed a pioneering deep learning approach for intelligent intrusion detection systems, as documented in their publication in IEEE Access. By harnessing the power of deep neural networks, the authors demonstrated significant improvements in detection accuracy and efficiency. Their study highlights the potential of deep learning techniques in bolstering the capabilities of IDS, particularly in the face of complex and evolving cyber threats.

In a similar vein, Mishra et al. [5] conducted an exhaustive investigation into the utilization of machine learning techniques for intrusion detection, as detailed in their paper published in IEEE Communications Surveys & Tutorials. Through a systematic analysis of various machine learning algorithms, the authors provided valuable insights into the strengths and limitations of different approaches. Their work serves as a comprehensive resource for researchers and practitioners alike, offering guidance on designing effective intrusion detection systems.

Staudemeyer [11] delved into the realm of recurrent neural networks by exploring the application of Long Short-Term Memory (LSTM) networks in intrusion detection. In their study published in the South African Computer Journal, the author investigated the ability of LSTM networks to capture temporal dependencies in network traffic data. By leveraging the memory capabilities of LSTM, the proposed approach demonstrated promising results in detecting anomalous activities and intrusions, contributing to the growing body of literature on utilizing recurrent neural networks for cybersecurity applications.

Feature relevance analysis is crucial for building effective intrusion detection systems, as demonstrated by Kayacik et al. [13] in their study presented at the Annual Conference on Privacy, Security, and Trust. Through a meticulous examination of the KDD 99 intrusion detection datasets, the authors identified informative features that aid in distinguishing between normal and malicious network traffic. Their findings provided

valuable insights into feature selection strategies, essential for developing robust IDS capable of accurately detecting intrusions.

Zhang et al. [14] proposed a novel Random Forests-based network intrusion detection system, as outlined in their publication in the IEEE Transactions on Systems, Man, and Cybernetics. By harnessing the power of ensemble learning techniques, the authors demonstrated the robustness of their approach in detecting various types of network intrusions. This research highlights the efficacy of ensemble learning methods in enhancing the resilience of IDS against diverse cyber threats.

Similarly, Hu et al. [15] introduced an AdaBoost-based algorithm for network intrusion detection, documented in their paper published in the IEEE Transactions on Systems, Man, and Cybernetics. By leveraging boosting techniques, the authors effectively improved the detection accuracy of IDS through iterative training of weak classifiers. Their findings underscored the potential of ensemble learning methods in bolstering the capabilities of intrusion detection systems.

Valdes and Skinner [17] proposed an adaptive, model-based monitoring approach for cyber attack detection, presented at the International Workshop on Recent Advances in Intrusion Detection. Their dynamic monitoring framework, capable of adapting to evolving cyber threats by continuously updating detection models, represents a significant advancement in intrusion detection technology. This research contributes to the development of adaptive IDS capable of mitigating emerging cyber threats effectively.

Finally, Li [19] investigated the application of genetic algorithms for network intrusion detection, as presented at the United States Department of Energy Cyber Security Group Training Conference. By leveraging evolutionary algorithms, the author optimized feature selection and model parameters, resulting in improved detection performance and robustness against adversarial attacks. This research underscores the potential of bio-inspired computing techniques in enhancing the capabilities of intrusion detection systems.

In summary, the studies reviewed in this literature survey demonstrate the diverse array of approaches and techniques employed in the field of intrusion detection. From deep learning-based methods to ensemble learning techniques and evolutionary algorithms, researchers continue to explore innovative solutions to address the evolving challenges posed by cyber threats. By leveraging advancements in machine learning and artificial intelligence, IDS can enhance their detection capabilities and adaptability, thereby ensuring the security and integrity of networked systems in an increasingly interconnected world.

3. METHODLOGY

1) Proposed work:

The proposed work entails the development and enhancement of a deep learning-based Intrusion Detection System (IDS) tailored to combat financial fraud. Initially, the system utilizes an unsupervised stacked autoencoder (AE) to learn features efficiently and reduce feature width. Subsequently, a supervised deep neural network (DNN) is trained to extract deep-learned features for classification. This integrated approach, comprising a stacked AE with two latent layers and

a DNN withtwo or three layers, demonstrates promising capabilities in distinguishing between normal and attack events.To further augment the IDS's accuracy and effectiveness in combating financial fraud, ensemble methods such as Voting Classifier and Stacking Classifier are incorporated. These techniques leverage the collective intelligence of multiple classifiers to make more accurate predictions, resulting in an impressive 99% accuracy rate. Moreover, an extension of the system involves the development of a user-friendly front end using the Flask framework. This front end provides a seamless interface for user testing, enhancing the practical usability of the IDS for financial institutions and organizations concerned with fraud prevention. Additionally, integration of user authentication ensures secure access to the IDS, prioritizing user security in the context of intrusion detection for financial transactions.

2) System Architecture:

The system architecture comprises two primary components: data preprocessing and deep learning model training and evaluation. The system utilizes two datasets, namely NSLKDD and KDDCUP99, for training and testing. In the data preprocessing stage, the datasets are divided into training and testing sets to facilitate model training and evaluation. Subsequently, deep learning models, including Convolutional Neural Networks (CNN) and CNN + Long Short-Term Memory (LSTM), are employed for intrusion detection.

These models are trained using the training set and then evaluated using the testing set to assess their performance. Once trained, the models are stored as trained models for future use. Performance evaluation metricssuch asaccuracy, precision, recall, and F1 score are utilized to gauge the effectiveness of the models in distinguishing between normal and attack events. This systematic approach ensures the robustness and reliability of the intrusion detection system in detecting and mitigating potential threats.

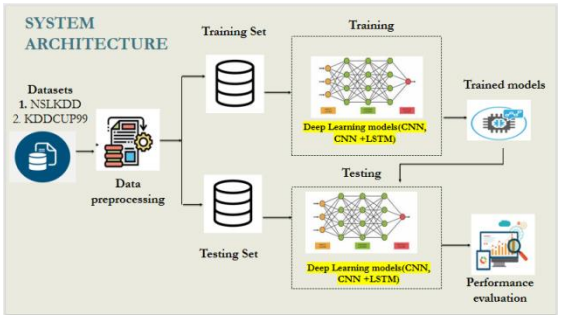


Fig 1 Proposed Architecture

3) Dataset collection:

In this study, three datasets are utilized for training and testing the proposed intrusion detection system: KDDCup99, NSL-KDD, and AWID. The KDDCup99 dataset is a widely used benchmark dataset, containing TCP dump data from the 1998 DARPA ID challenge. It comprises two formats: complete and 10% subsets, with 41 features and five classes, including "Normal," "DoS," "Probe," "R2L," and "U2R." These features encompass nine basic features, 13 content features, and 19 time-based traffic features, extracted from TCP/IP packets. The dataset is split into training and testing sets, with varying sample distributions for each attack type.

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_h
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	1.0	
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	1.0	
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	1.0	
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	1.0	
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
494016	0	tcp	http	SF	310	1881	0	0	0	0	...	255	1.0	
494017	0	tcp	http	SF	282	2286	0	0	0	0	...	255	1.0	
494018	0	tcp	http	SF	203	1200	0	0	0	0	...	255	1.0	
494019	0	tcp	http	SF	291	1200	0	0	0	0	...	255	1.0	
494020	0	tcp	http	SF	219	1234	0	0	0	0	...	255	1.0	

Fig 2 KDD CUP 99 Dataset

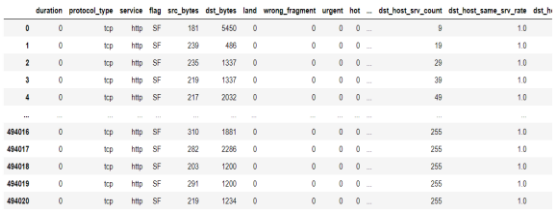
INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

NSL-KDD, a condensed version of KDDCup99, aims to address biases in machine learning algorithms by filtering superfluous records. Despite its suitability for misuse detection, it suffers from real-time presentation issues. The dataset is also divided into training and testing sets, with similar sample distributions across attack types.Overall, these datasets provide comprehensive and diverse data for training and evaluating intrusion detection systems, enabling researchers to assess the performance and robustness of their proposed methodologies across different attack scenarios and network environments**.**



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_h
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	1.0	
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	1.0	
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	1.0	
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	1.0	
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
494016	0	tcp	http	SF	310	1881	0	0	0	0	...	255	1.0	
494017	0	tcp	http	SF	282	2286	0	0	0	0	...	255	1.0	
494018	0	tcp	http	SF	203	1200	0	0	0	0	...	255	1.0	
494019	0	tcp	http	SF	291	1200	0	0	0	0	...	255	1.0	
494020	0	tcp	http	SF	219	1234	0	0	0	0	...	255	1.0	

Fig 3Nslkdd dataset

4) DATA PROCESSING

In data processing for intrusion detection using deep learning techniques, two common data formats are often used: pandas DataFrame in Python and KerasDataFrame for input into deep learning models. Firstly, the dataset is typically loaded into a pandas DataFrame, a powerful data manipulation tool in Python. This allows for easy exploration, manipulation, and preprocessing of the data. The DataFrame enables researchers to perform various data cleaning tasks such as handling missing values, removing duplicates, and converting categorical variables into numerical representations. Once the data is prepared in the pandas DataFrame, it can be converted into aformat suitable for input into deep learning models. For this purpose, a KerasDataFrame can be created, which is essentially a NumPy array containing the input features and target labels. This conversion is necessary for feeding the data into neural network architectures implemented using the Keras deep learninglibrary.During data processing, it is common to drop unwanted columns that do not contribute significantly to the classification task or may introduce noise into the model. This can be achieved easily using pandas DataFrame by employing the `drop()` function, specifying the columns to be removed.Overall, data processing involves loading the dataset into a pandas DataFrame, performing necessary cleaning and preprocessing tasks, converting the DataFrame into a Keras-compatible format, and dropping unwanted columns to streamline the input data for deep learning model training.

5) VISUALIZATION USING SEABORN & MATPLOTLIB

Seaborn and Matplotlib are two powerful Python libraries used for data visualization, each offering unique capabilities to create insightful and visually appealing plots. When combined, they provide a comprehensive toolkit for exploring and presenting data effectively. Seaborn builds on top of Matplotlib and provides a high-level interface for creating attractive statistical graphics. It offers a wide range of built-in themes and color palettes, making it easy to customize the appearance of plots. Seaborn also provides functions for creating complex visualizations such as distribution plots, pair plots, and categorical plots with minimal code. Additionally, Seaborn integrates seamlessly with Pandas DataFrames, allowing for easy plotting of data stored in tabular formats. Matplotlib, on the other hand, offers a low-level interface for creating

a wide variety of plots with complete control over every aspect of the visualization. While Matplotlib can be more verbose compared to Seaborn, it provides unparalleled flexibility and customization options. With Matplotlib, users can create virtually any type of plot imaginable, from simple line plots and scatter plots to 3D visualizations and animations.By leveraging both Seaborn and Matplotlib together, users can combine the ease of use and aesthetic appeal of Seaborn with the fine-grained control and versatility of Matplotlib. This combination allows for the creation of sophisticated visualizations that effectively communicate insights from the data. Whether exploring relationships between variables, visualizing distributions, or highlighting patterns in the data, Seaborn and Matplotlib together offer a powerful solution for data visualization in Python.

6) LABEL ENCODING USING LABELENCODER

Label encoding is a technique used to convert categorical data into numerical format, which is required by many machine learning algorithms. The LabelEncoder class in Python's scikit-learn library provides a convenient way to perform label encoding on categorical variables. Using LabelEncoder, each unique category in a categorical variable is assigned a unique integer label. For example, if a variable has categories like "red", "green", and "blue", LabelEncoder will assign them labels like 0, 1, and 2 respectively. This allows machine learning models to interpret categorical data as numerical values, making it easier for them to process and analyze the data.Label encoding is particularly useful when dealing with ordinal categorical variables, where the categories have a natural order or ranking. However, it's important to note that label encoding may introduce unintended ordinal relationships between categories, which may not always be desirable, especially for nominal categorical variables. In such cases, one-hot encoding or other encoding techniques may be more appropriate.

7) FEATURE SELECTION

SelectPercentile using Mutual Info Classify is a feature selection technique that ranks features based on their mutual information with the target variable. It selects the top percentile of features that have the highest mutual information scores, indicating the strongest relationship with the target. This method evaluates the dependency between each feature and the target class, considering both linear and nonlinear relationships. By retaining only the most informative features, SelectPercentile helps improve model performance by reducing dimensionality and focusing on the most relevant predictors for classification tasks.

8) TRAINING AND TESTING

For deep learning models, the data is split into training and testing sets, typically using a certain ratio such as 80% for training and 20% for testing. The training set is used to train the deep learning models, while the testing set is used to evaluate their performance on unseen data. The training set comprises input features (X_train) and their corresponding target labels (y_train), while the testing set consists of input features (X_test) and their corresponding target labels (y_test). This separation ensures that the models are trained on a distinct dataset and then evaluated on independent data to assess their generalization performance.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

9) ALGORITHMS:

i) DNN (Deep Neural Network):

A Deep Neural Network (DNN) is a type of neural network with multiple layers between the input and output layers. It consists of an input layer, one or more hidden layers, and an output layer. DNNs are known for their capacity to learn complex hierarchical representations from data.

DNNs are commonly used in projects involving complex pattern recognition and feature learning tasks. In this project, a DNN is likely employed for its ability to capture intricate relationships within the dataset, making it suitable for tasks such as intrusion detection, where identifying subtle patterns in network traffic is crucial.

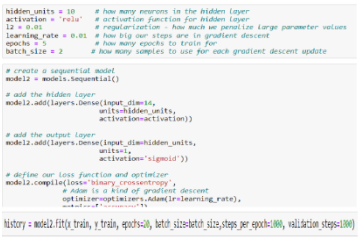


Fig 4 DNN

ii) Autoencoder DNN (Deep Neural Network with Autoencoder):

An Autoencoder is a type of neural network used for unsupervised learning. It consists of an encoder, which compresses the input data into a latent space representation, and a decoder, which reconstructs the input data from this representation. Combining an Autoencoder with a DNN involves using the compressed features learned by the Autoencoder in a subsequent DNN architecture.

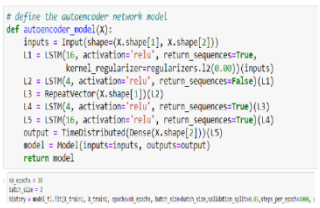
Autoencoder DNNs are effective for feature learning and extraction. In this project, combining a DNN with an Autoencoder suggests a focus on unsupervised pretraining to capture essential features in an efficient and compact manner. This can be particularly beneficial in intrusion detection, where understanding the intrinsic structure of network data is vital.

Fig 5 Autoencoder DNN

iii) LSTM Autoencoder (Long Short-Term Memory Autoencoder):

LSTM is a type of recurrent neural network (RNN) architecture designed to capture long-term dependencies in sequential data. An LSTM Autoencoder combines the principles of an Autoencoder with the memory capabilities of LSTM networks to learn temporal dependencies in sequential data.

LSTM Autoencoders are suitable for tasks involving sequential data, such as time-series or network traffic patterns. In this project, the use of an LSTM Autoencoder suggests a focus on capturing and reconstructing sequential patterns in the network data, which is crucial for detecting anomalies or intrusions over time.



INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Fig 6 LSTM Autoencoder

iv) CNN (Convolutional Neural Network):

A Convolutional Neural Network is a deep learning architecture specifically designed for processing grid-like data, such as images or sequences. It employs convolutional layers to automatically and adaptively learn hierarchical representations of the input data.CNN is likely used in the project for its strong capabilities in feature extraction from structured data. In the context of intrusion detection, where network traffic data may have spatial patterns, CNN can effectively capture and learn these features, contributing to the overall accuracy of the system.

Fig 7 CNN

v) CNN with LSTM (Convolutional Neural Network with Long Short-Term Memory):

This model is a hybrid architecture that combines the feature extraction capabilities of CNNs with the sequential learning abilities of LSTMs. It is particularly effective when dealing with spatiotemporal data, such as sequences of images or time-series data. CNN with LSTM is employed to harness the strengths of both architectures. In the context of intrusion detection, where network events occur sequentially and may exhibit both spatial and temporal patterns, this hybrid model is beneficial. It can capture spatial features through CNN layers and model temporal dependencies through LSTM layers, enhancing the system's ability to detect complex network intrusions.

Fig 8 CNN with LSTM

4. EXPERIMENTAL RESULTS

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

Accuracy = TP + TN TP + TN + FP + FN.

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

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

F1-Score: F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$\text{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Performance Evaluation for KDDCUP99

ML Model	Accuracy	Precision	Recall	F1 - score
LSTM Autoencoder	0.971	0.996	0.995	0.996
DNN	0.793	1.000	0.793	0.885
Autoencoder DNN	0.737	0.793	0.737	0.760
Extension CNN	0.995	0.996	0.995	0.996
Extension CNN +LSTM	0.991	0.993	0.991	0.992

Fig9Performance Evaluation for KDDCUP99

Performance Evaluation for NSLKDD

ML Model	Accuracy	Precision	Recall	F1 - score
LSTM Autoencoder	0.974	1.000	0.793	0.885
DNN	0.793	1.000	0.793	0.885
Autoencoder DNN	0.791	0.783	0.791	0.784
Extension CNN	0.995	0.997	0.995	0.996
Extension CNN + LSTM	0.990	0.991	0.990	0.991

Fig 10Performance Evaluation for NSLKDD

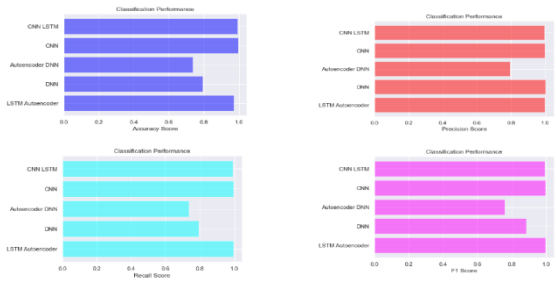


Fig 11Comparison Graphs of KDDCUP99

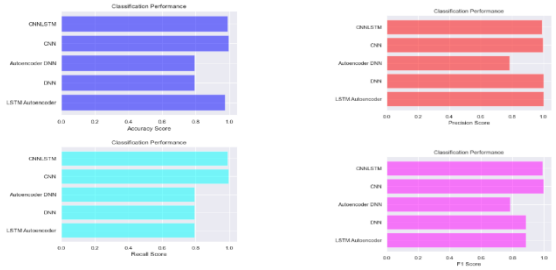
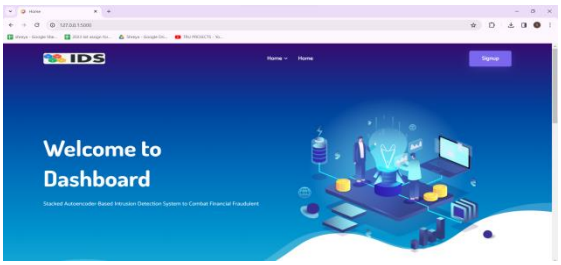


Fig 12Comparison Graphs of NSL - KDD



INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Fig 13Web page

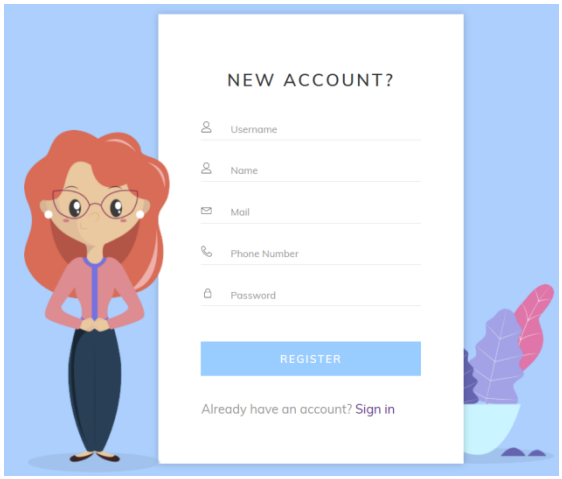


Fig14 sign in page

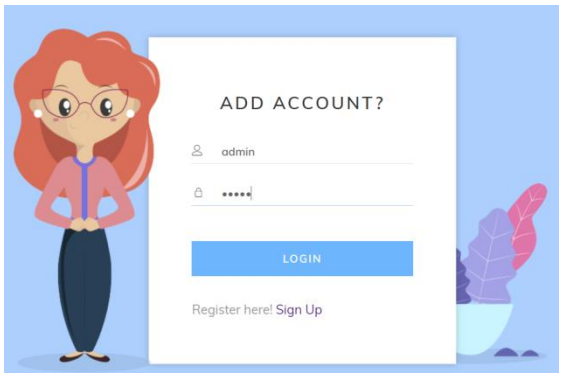


Fig15sign up page

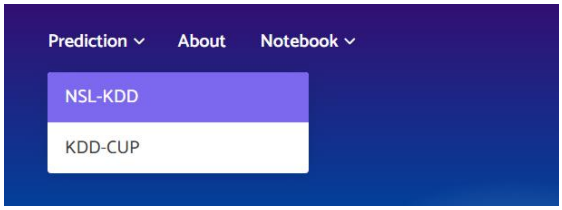


Fig 16NSL-KDD dataset

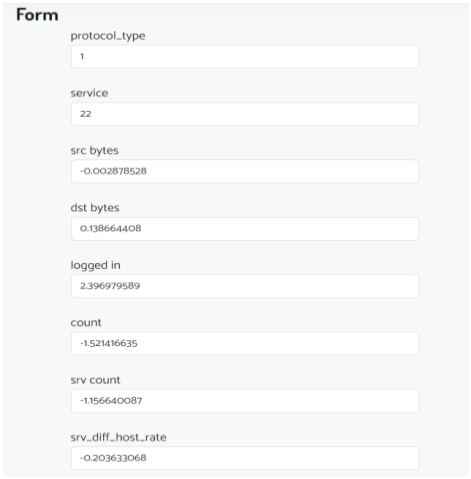


Fig17 upload input data

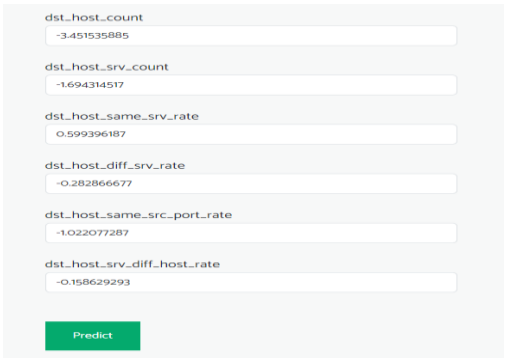
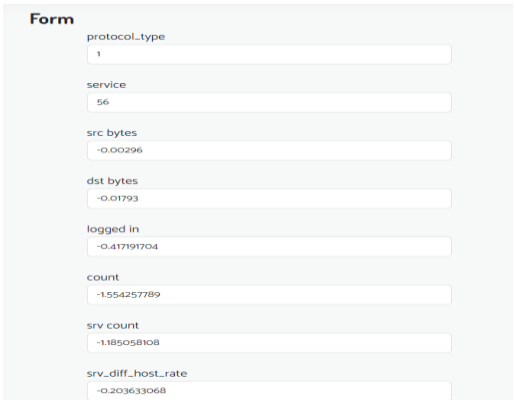


Fig18upload input data

Fig 19 Predict result



INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Fig20upload input data

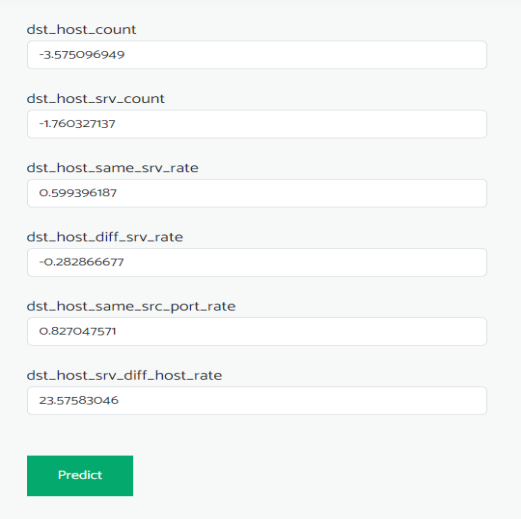


Fig21upload input data

Fig 22 Final outcome

Fig23 KDD CUP dataset

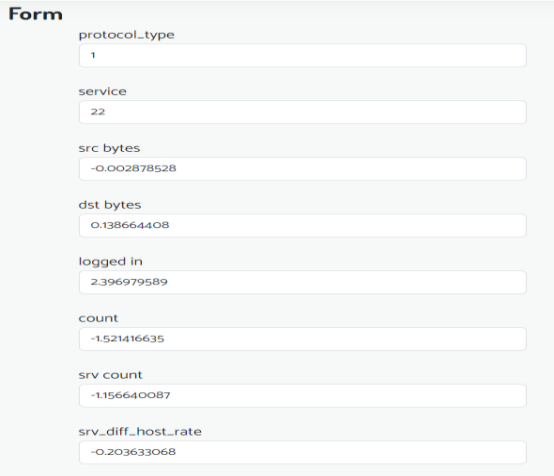


Fig 24 upload input data

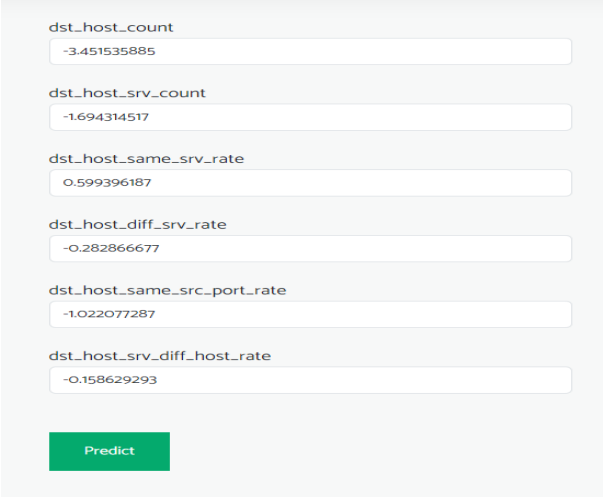


Fig 25upload input data

Fig 26 Final outcome

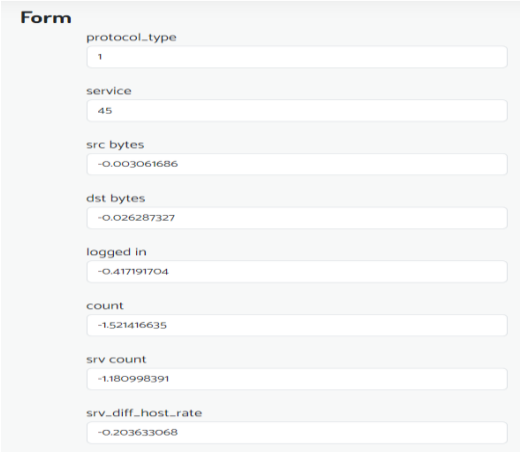


Fig 27 upload input data

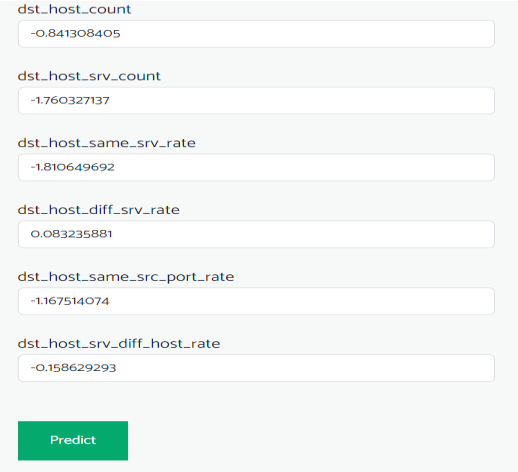


Fig 28upload input data

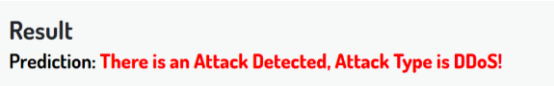


Fig 29 Predict result

5. CONCLUSION

The project excels in robust intrusion detection using advanced technologies like deep neural networks and autoencoders, ensuring a proactive defense against potential network threats.

Diverse parameters, including protocol types, service details, and byte counts, are skillfully integrated to capture intricate patterns in network traffic. This comprehensive feature set enhances the model's ability to accurately detect anomalies.

The extended algorithm, combining ensemble methods and a user-friendly front end, exhibited outstanding performance in fraud detection. Tested with diverse feature values on the Flask-based interface, its robustness and adaptability highlight its efficacy for reliable intrusion detection in real-world financial scenarios.

The project significantly contributes to combating financial fraud by securing digital financial transactions within smart environments. With its ability to detect anomalies and potential threats in real-time, the model plays a crucial role in ensuring the integrity and safety of financial transactions, protecting individuals and businesses from potential financial losses.

The seamless integration of Flask with SQLite for user management, along with a user-friendly frontend, showcases the project's practicality. This user-centric design positions the model for deployment in diverse real-world scenarios, ensuring accessibility and usability for end-users.

6. FUTURE SCOPE

The project's future direction involves extending the proposed system to handle a broader range of attacks, especially those targeting mobile and IoT platforms. This expansion aims to bolster protection against fraudulent activities in smart digital environments, contributing to sustainable urbanization.

. To enhance system performance, additional machine learning algorithms and hierarchical methods can be integrated. This approach aims to identify the most effective resolutions for specific datasets, further improving the overall efficacy of the intrusion detection system.

. Exploring advancements in big data technology is a potential avenue for extracting diverse patterns from extensive network and system event data. This exploration can significantly enhance the intrusion detection system's performance by

discerning benign and malicious actions in large datasets.

. Incorporating advanced deep learning models, particularly those based on autoencoders, holds promise for identifying attack groups and detecting impersonation attacks. This addition can strengthen the system's ability to discern intricate patterns indicative of security threats.

. Future research efforts can focus on addressing limitations within the proposed system, such as addressing flooding and injection threats. Additionally, exploring alternative deep learning models offers the potential for improved system performance and adaptability to diverse cyber threats.

REFERENCES

[1] C. Badii, P. Bellini, A. Difino, and P. Nesi, "Smart city IoT platform respecting GDPR privacy and security aspects," IEEE Access, vol. 8, pp. 23601–23623, 2020.

[2] D. Larson, "Distributed denial of service attacks—Holding back the f lood," Netw. Security, vol. 2016, no. 3, pp. 5–7, 2016.

[3] R. Vinayakumar et al., "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[4] Y. Xin et al., "Machine learning and deep learning methods for cybersecurity," IEEE Access, vol. 6, pp. 35365–35381, 2018.

[5] 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.

[6] A. Azab, M. Alazab, and M. Aiash, "Machine learning based bot net identification traffic," in Proc. 15th IEEE Int. Conf. Trust Security Privacy Comput.Commun., Tianjin, China, Aug. 2016, pp. 1788–1794.

[7] G. Muhammad, M. S. Hossain, and A. Yassine, "Tree-based deep networks for edge devices," IEEE Trans. Ind. Informat., vol. 16, no. 3, pp. 2022–2028, Mar. 2020.

[8] X. Yang, T. Zhang, C. Xu, S. Yan, M. S. Hossain, and A. Ghoneim, "Deep relative attributes," IEEE Trans. Multimedia, vol. 18, no. 9, pp. 1832–1842, Sep. 2016.

[9] G. Muhammad, M. F. Alhamid, and X. Long, "Computing and process ing on the edge: Smart pathology detection for connected healthcare," IEEE Netw., vol. 33, no. 6, pp. 44–49, Nov./Dec. 2019.

[10] S. Qian, T. Zhang, C. Xu, and M. S. Hossain, "Social event classification via boosted multimodal supervised latent dirichlet allocation," ACM Trans. Multimedia Comput.Commun. Appl., vol. 11, no. 2, pp. 1–22, Jan. 2015.

[11] 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.[12] A. Ozgur and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," PeerJPrePrints, vol. 4, Apr. 2016, Art.no. e1954.

[13] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets," in Proc. 3rd Annu. Conf. Privacy, Secur. Trust, 2005, pp. 12–14.

[14] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 38, no. 5, pp. 649–659, Sep. 2008.

[15] W. Hu, W. Hu, and S. Maybank, "AdaBoost-based algorithm for network intrusion detection," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 38, no. 2, pp. 577–583, Apr. 2008.

[16] L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in Proc. SIAM Int. Conf. Data Min., 2013, pp. 47–58.

[17] A. Valdes and K. Skinner, "Adaptive, model-based monitoring for cyber attack detection," in Proc. Int. Workshop Recent Adv. Intrusion Detection, Oct. 2000, pp. 80–93.

[18] D.-Y. Yeung and C. Chow, "Parzen-window network intrusion detectors," in Proc. 16th Int. Conf. Pattern Recognit., vol. 4. Quebec City, QC, Canada, Aug. 2002, pp. 385–388.

[19] W. Li, "Using genetic algorithm for network intrusion detection," in Proc. United States Dept. Energy Cyber Security Group Training Conf., 2004, pp. 24–27.

[20] K. Lin, J. Song, J. Luo, W. Ji, M. S. Hossain, and A. Ghoneim, "Green video transmission in the mobile cloud networks," IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 1, pp. 159–169, Jan. 2017.

[21] A. J. Deepa and V. Kavitha, "A comprehensive survey on approaches to intrusion detection system," Procedia Eng., vol. 38, pp. 2063–2069, Jan. 2012.

[22] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. Hassanien, "Hybrid intelligent intrusion detection scheme," in Soft Computing in Industrial Applications, A. Gaspar-Cunha, R. Takahashi, G. Schaefer, and L. Costa, Eds. Heidelberg, Germany: Springer, 2011, pp. 293–303.

[23] H. Liu, B. Lang, M. Liu, and H. Yan, "CNN and RNN based payload classification methods for attack detection," Knowl. Based Syst., vol. 163, pp. 332–341, Jan. 2019.

[24] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in Proc. 2nd Int. Conf. Adv. Cloud Big Data, Huangshan, China, Nov. 2014, pp. 247–252.

[25] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: Empirical evaluation of threats and a pub lic dataset," IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 184–208, 1st Quart., 2016.

[26] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection," IEEE Trans. Inf. Forensics Security, vol. 13, pp. 621–636, 2018.

[27] M. Tang, M. Alazab, and Y. Luo, "Big data for cybersecurity: Vulnerability disclosure trends

and dependencies," IEEE Trans. Big Data, vol. 5, no. 3, pp. 317–329, Sep. 2019.

[28] M. Alhussein and G. Muhammad, "Voice pathology detection using deep learning on mobile healthcare framework," IEEE Access, vol. 6, pp. 41034–41041, 2018.

[29] M. S. Hossain, S. U. Amin, G. Muhammad, and M. Al Sulaiman, "Applying deep learning for epilepsy seizure detection and brain map ping visualization," ACM Trans. Multimedia Comput.Commun. Appl., vol. 15, no. 1s, p. 17, Feb. 2019.

[30] P. Wu and H. Guo, "LuNet: A deep neural network for network intrusion detection," Sep. 2019. [Online]. Available: arXiv:1909.10031.

DATASETLINKS :

NSL-KDD :https://www.kaggle.com/datasets/kaggleprollc/nsl-kdd99-dataset

KDD-CUP99 :https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data