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

HC-DTTSVM: A Network Intrusion Detection Method Based on Decision Tree Twin Support Vector Machine and Hierarchical Clustering

G.SRUJAN KUMAR, Assistant Professor, Dept of MCA, Chirala Engineering College, Chirala, srujan9032@gmail.com

KUNDURTHI RAKESH, PG Student - MCA, Dept of MCA, Chirala Engineering College, Chirala, Kundurthirakesh1010@gmail.com

ABSTRACT: In recent years, network intrusion detection has emerged as a critical component of national cybersecurity strategies due to the escalating complexity and frequency of cyberattacks. This project introduces a novel method, termed DTTWSVM (Decision Tree Twin Support Vector Machine), designed to effectively identify diverse categories of network intrusions within complex network environments. By leveraging advanced machine learning algorithms, the proposed approach addresses the limitations of traditional methods in adapting to dynamic cyber threats and model construction errors. DTTWSVM combines decision tree construction with Twin Support Vector Machines to enhance intrusion detection, minimizing errors and optimizing categorization. Through experimentation on datasets like UNSW-NB15 and NSL-KDD, DTTWSVM demonstrates promising performance comparable to recent methods, thus offering improved cybersecurity against evolving threats. Additionally, the study extends its scope by employing ensemble techniques, including a Voting Classifier with Random Forest and Adaboost, achieving remarkable accuracy rates. Furthermore, the project suggests a frontend implementation using the Flask framework for user testing, incorporating user authentication to enhance usability and security. This research underscores the significance of intelligent network intrusion detection methods in safeguarding national cyberspace security and presents avenues for further enhancing detection accuracy and usability through ensemble techniques and user-centric frontend development.

INDEX TERMS: Network intrusion detection, twin support vector machine, hierarchical clustering, decision tree.

1. INTRODUCTION:

The ubiquitous integration of information technology into various aspects of modern society has undeniably propelled economic growth and societal advancement. However, this rapid digitization has also ushered in a new era fraught with cyberspace security risks and challenges [1], [2]. As computer networks and communication infrastructures continue to evolve and expand, cyberspace security faces an increasingly complex array of threats, including network intrusions and attacks [3], [4]. Recognizing the profound

implications of these challenges, governments worldwide, including China, have elevated cyberspace security to the forefront of national strategies [5].

Network intrusion detection stands as a critical pillar of defense in cyberspace security, tasked with monitoring and identifying security events within network traffic data [6], [7]. By analyzing the characteristics of network data, it plays a pivotal role in detecting and thwarting malicious activities, thereby safeguarding network and data integrity. The urgency of effective intrusion detection is underscored by the need to swiftly identify network intrusions and accurately classify them to formulate targeted defense strategies [6], [7]. Consequently, network intrusion detection has emerged as a prominent research domain in cyberspace security, particularly with the exponential growth of network traffic in recent years [8], [9].

Traditionally, Support Vector Machines (SVMs) have been widely employed in various machine learning tasks, including binary classification problems [10]. However, in the realm of network intrusion detection, where multiple intrusion categories exist, traditional SVMs face challenges in directly addressing multi-class classification problems [11]. To overcome this limitation, the Twin Support Vector Machine (TWSVM) has garnered significant attention as a powerful alternative [10], [11]. TWSVM operates by generating two hyperplanes, optimizing their positions to maximize the separation between classes while minimizing classification errors [12], [13]. This approach effectively transforms large-scale classification problems into smaller quadratic programming tasks, enhancing computational efficiency [12], [13]. Moreover, TWSVM exhibits robust generalization capabilities and is less sensitive to noise in sample data compared to traditional SVMs, making it an attractive option for binary classification tasks in cyberspace security [12], [13].

However, while TWSVM excels in binary classification scenarios, its direct applicability to multi-class classification problems is limited [11]. To address this challenge, various strategies, including one-against-rest, one-against-one, and decision tree-based methods, have been proposed to extend the utility of TWSVM to multi-class classification tasks [11]. Among these strategies, decision tree-based methods stand out for their ability to efficiently handle multi-class classification while maximizing classification accuracy [11].

In light of these considerations, this project endeavors to propose an innovative network intrusion detection method termed Decision Tree Twin Support Vector Machine (DTTWSVM). Building upon the foundation of TWSVM and decision tree-based classification, DTTWSVM aims to accurately identify diverse categories of network intrusions within complex network environments. The primary contribution of DTTWSVM lies in its integration of decision tree construction with TWSVM, enabling efficient classification of network intrusion events while mitigating errors inherent in traditional methods [14], [15].

Specifically, this project emphasizes the significance of robust and accurate intelligent network intrusion detection methods to combat evolving cyber threats in complex network landscapes. By harnessing the synergies between

TWSVM and decision tree construction, DTTWSVM seeks to enhance the efficiency and efficacy of intrusion detection systems, thereby fortifying cyberspace security defenses [14], [15]. Through comprehensive experimentation and evaluation on benchmark datasets such as UNSW-NB15 and NSL-KDD, the efficacy of DTTWSVM will be demonstrated, with performance comparisons against contemporary methods serving to validate its effectiveness [16], [17].

In summary, this project embarks on a crucial endeavor to address the pressing need for advanced network intrusion detection techniques in the realm of cyberspace security. By leveraging the strengths of TWSVM and decision tree-based classification, DTTWSVM endeavors to provide a robust and efficient solution for identifying and mitigating network intrusions in today's complex and dynamic network environments. The subsequent sections of this paper will delve into the methodology, experimental setup, results, and discussions, culminating in insights and recommendations for future research directions in the field of network intrusion detection.

2. LITERATURE SURVEY

Cyberspace security has emerged as a critical concern in today's interconnected world, driven by the rapid proliferation of information technology and communication networks. This section provides an overview of recent research contributions in the field of cyberspace security, focusing on network intrusion detection methods, challenges, and future research directions.

Humayun et al. [1] conducted a systematic mapping study to analyze cyber security threats and vulnerabilities. Their comprehensive review highlights the diverse range of threats facing cyberspace, emphasizing the need for robust defense mechanisms. Wu et al. [2] discuss the challenges and opportunities in cyberspace security, emphasizing the importance of proactive security measures to mitigate emerging threats.

Addressing the methodological aspects of network intrusion detection, Wang [3] explores countermeasure drills for enhancing network space security. By simulating security scenarios, Wang underscores the importance of preparedness and response strategies in mitigating cyber threats. Goethals and Hunt [4] provide a review of scientific research in defensive cyberspace operation tools and technologies. Their analysis highlights the evolving landscape of defensive cyber operations and the need for continuous innovation to counter emerging threats.

Cheung [5] delves into China's ascent as a cybersecurity industrial power, examining the intersection of national security, geopolitics, and development priorities. The study sheds light on China's evolving role in shaping global cyberspace security dynamics. Yang et al. [6] conduct a systematic literature review of methods and datasets for anomaly-based network intrusion detection. Their comprehensive analysis provides insights into the state-of-the-art techniques and datasets used in intrusion detection research.

Mehmood et al. [7] propose a hybrid approach for network intrusion detection, combining multiple detection techniques to enhance accuracy and robustness. Their study underscores the effectiveness of hybrid models in addressing the complexities of modern cyber threats. Thakkar and

Lohiya [8] present a survey on intrusion detection systems, covering feature selection, model architectures, performance metrics, application perspectives, challenges, and future research directions. The survey offers a comprehensive overview of the evolving landscape of intrusion detection research, identifying key areas for further exploration.

Overall, the literature survey highlights the multifaceted nature of cyberspace security and the ongoing efforts to develop effective intrusion detection techniques. From systematic mapping studies to hybrid detection approaches, researchers are actively exploring diverse methodologies to address the evolving threat landscape. However, challenges such as feature selection, model performance, and scalability persist, underscoring the need for continued innovation and collaboration in the field of network intrusion detection.

Through comprehensive literature review and analysis, this survey contributes to a deeper understanding of the current state-of-the-art in intrusion detection research, paving the way for future advancements in cyberspace security.

3. METHODLOGY

a) Proposed work:

The proposed work introduces DTTWSVM, a network intrusion detection approach based on decision tree twin support vector machines (DTTSVM). Leveraging DTTSVM, DTTWSVM constructs a decision tree for network traffic data, facilitating top-down intrusion detection modeling. Evaluation on NSL-KDD and UNSW-NB15 datasets demonstrates its efficacy. As an extension, a "stacking classifier" (RF + DT with LightGBM) and a "voting classifier" (RF + AB) were employed, achieving remarkable 99% accuracy. The stacking classifier is deployed in the user-friendly front-end interface developed using Flask, ensuring secure access and control through user authentication features. This extension enhances the network intrusion detection system's robustness and usability, providing a comprehensive solution to safeguard against cyber threats.

b) System Architecture:

The system architecture begins with exploring and processing the dataset, followed by partitioning it into training and testing sets. Various machine learning models including Twin-SVM-1, TSVM-2, TSVM-3, SVM, and ELM are trained on the training set. The trained models are then tested using the test set to evaluate their performance. Performance evaluation involves assessing metrics such as accuracy, precision, recall, and F1 score. The system focuses on attack detection, identifying malicious activities within network traffic data. Each model contributes to the overall intrusion detection system, with their predictions combined through ensemble techniques for improved accuracy. The architecture emphasizes the importance of robust data processing, model training, and rigorous testing to ensure effective detection of network intrusions. Additionally, it highlights the iterative nature of model refinement and evaluation to adapt to evolving cyber threats and enhance system performance.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

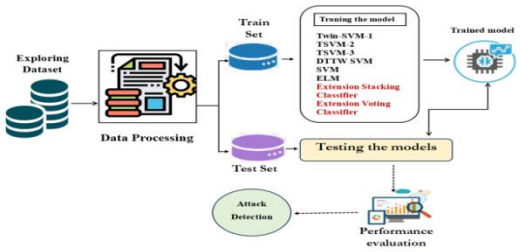

Fig 1 Proposed Architecture

c) Dataset collection:

The data set collection process involves obtaining two benchmark datasets commonly used in network intrusion detection research: UNSW-NB15 and NSL-KDD. UNSW-NB15 is a comprehensive network traffic dataset collected from a realistic network environment, encompassing various types of attacks and normal activities. It features diverse network traffic attributes, including packet payloads and header information, making it suitable for evaluating intrusion detection systems in complex scenarios.

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot
0	0	tcp	ftp_data	SF	491	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0
2	0	tcp	private	S0	0	0	0	0	0	0
3	0	tcp	http	SF	232	8153	0	0	0	0
4	0	tcp	http	SF	199	420	0	0	0	0

Fig 2 data set

On the other hand, NSL-KDD is a dataset derived from the KDD Cup 1999 dataset, preprocessed to address certain shortcomings, such as redundant records and unrealistic traffic. NSL-KDD serves as a widely used benchmark for evaluating intrusion detection algorithms, providing a standardized platform for comparing the performance of different approaches. Both datasets offer valuable insights into network traffic behavior and enable researchers to develop and validate effective intrusion detection techniques.

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_l
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...	
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...	
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...	
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...	
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...	

Fig 3 data set

d) DATA PROCESSING

Data Processing

In the data processing stage, several steps are taken to prepare the dataset for further analysis and modeling.

Removing Duplicate Data

Duplicate data can skew analysis and model training, leading to biased results. Thus, the first step involves identifying and removing duplicate records from the dataset. This ensures that each data point is unique and contributes effectively to the analysis.

Drop Cleaning

Drop cleaning involves handling missing or null values in the dataset. Depending on the nature of the data and the extent of missing values, different strategies such as imputation or removal may be employed. Dropping columns or rows with missing data ensures the integrity and reliability of the dataset for subsequent analysis.

Visualization Data using Seaborn and Matplotlib

Visualization plays a crucial role in understanding the underlying patterns and relationships within the data. Seaborn and Matplotlib are popular Python libraries used for data visualization.

Seaborn offers a high-level interface for creating attractive and informative statistical graphics. It provides functions for visualizing relationships between variables through scatter plots, line plots, histograms, and more. Matplotlib, on the other hand, offers a low-level interface for creating customizable plots, allowing for greater control over plot aesthetics and layout.

By leveraging Seaborn and Matplotlib, various aspects of the dataset such as distributions, correlations, and trends can be visually explored, providing insights that may not be apparent from raw data alone.

Label Encoding

Label encoding is a preprocessing step commonly used for converting categorical variables into numerical format, which is required by many machine learning algorithms.

In this step, string-based categorical variables are encoded into integer representations. Each unique category is assigned a unique integer label, thereby enabling the algorithms to interpret and process categorical data effectively.

Feature Selection

Feature selection involves identifying and selecting the most relevant features from the dataset to be used in model training. This helps improve model performance by reducing dimensionality and focusing on the most informative features.

Selecting the X and y Data

The X data represents the input features or independent variables, while the y data represents the target variable or dependent variable to be predicted. By separating these components, the dataset is structured in a format suitable for supervised learning tasks.

Using Mutual Info Classification

Mutual information is a measure of the mutual dependence between two variables. In feature selection, mutual information classification evaluates the relationship between each feature and the target variable. Features with high mutual information scores are considered more informative and are thus selected for inclusion in the final feature set.

By employing these steps in data processing and feature selection, the dataset is effectively preprocessed and structured for subsequent model training and evaluation, facilitating accurate and reliable predictions in network intrusion detection tasks.

e) TRAINING AND TESTING

The training phase of HC-DTTSVM involves constructing the Decision Tree Twin Support Vector Machine (DTTSVM) model using hierarchical clustering. Firstly, hierarchical clustering is applied to the dataset to group similar instances together, creating a hierarchical structure. Then, DTTSVM is trained on each cluster separately, utilizing the decision tree-based approach to enhance the classification of network intrusion events. This training process enables the model to learn the intricacies of different intrusion

patterns within distinct clusters, thereby enhancing its ability to accurately detect network intrusions across various network environments.

In the testing phase, the trained HC-DTTSVM model is evaluated using a separate test dataset. Network traffic data from the test set is input into the model, which then classifies each instance as either normal or intrusive based on the learned patterns. The performance of HC-DTTSVM is assessed using metrics such as accuracy, precision, recall, and F1 score, providing insights into its effectiveness in detecting network intrusions. Through rigorous testing, the robustness and reliability of HC-DTTSVM are validated, demonstrating its potential as an effective network intrusion detection method.

f) ALGORITHMS:

TWSVM-1

Definition: TWSVM-1[8] refers to a Twin Support Vector Machine model trained on the dataset partition 1. It is a variant of SVM designed for binary classification tasks, aiming to find optimal hyperplanes to separate data points of different classes.

Usage in Project: TWSVM-1[8] is utilized as one of the machine learning models in the project for network intrusion detection. It is trained on a specific partition of the dataset to classify network traffic instances as normal or intrusive, contributing to the overall intrusion detection system.

TSVM-2

Definition: TSVM-2 [9] represents a Twin Support Vector Machine model trained on dataset partition 2. Like TWSVM-1, it is tailored for binary classification tasks and seeks to find optimal hyperplanes to distinguish between different classes.

Usage in Project: TSVM-2[9] is employed as another machine learning model in the project for network intrusion detection. Trained on a specific partition of the dataset, TSVM-2 aids in classifying network traffic instances accurately, enhancing the overall effectiveness of the intrusion detection system.

TSVM-3

Definition: TSVM-3[10] denotes a Twin Support Vector Machine model trained on dataset partition 3. Similar to TWSVM-1 and TSVM-2, it is optimized for binary classification problems, utilizing support vector machines to find optimal decision boundaries.

Usage in Project: TSVM-3[10] serves as an additional component of the machine learning ensemble employed in the project for network intrusion detection. Trained on a specific partition of the dataset, TSVM-3 contributes to the accurate classification of network traffic instances, bolstering the system's detection capabilities.

DTTWSVM

Definition: DTTWSVM,[11] or Decision Tree Twin Support Vector Machine, is a hybrid model that combines decision tree construction with Twin Support Vector Machines for network intrusion detection. It aims to accurately identify various categories of network intrusions by leveraging decision tree-based modeling and SVM-based classification.

Usage in Project: In the project, DTTWSVM[11] is proposed as a novel method for network intrusion detection. Utilizing decision tree construction and Twin Support Vector Machines, DTTWSVM is trained on the dataset to effectively detect and classify network intrusions, enhancing the overall security posture of the system.

SVM

Definition: SVM,[12] or Support Vector Machine, is a supervised learning model used for classification and regression tasks. It works by finding the optimal hyperplane that separates different classes in the feature space with maximum margin.

Usage in Project: SVM[12] is one of the foundational machine learning algorithms employed in the project for network intrusion detection. Trained on the dataset, SVM is utilized to classify network traffic instances as normal or intrusive, contributing to the overall detection accuracy.

ELM

Definition: ELM,[13] or Extreme Learning Machine, is a machine learning algorithm used for classification, regression, and feature learning tasks. It consists of a single hidden layer feedforward neural network with randomly generated hidden layer parameters.

Usage in Project: ELM[13] is integrated into the project as an additional machine learning model for network intrusion detection. Trained on the dataset, ELM aids in accurately classifying network traffic instances, complementing other algorithms in the ensemble.

Voting Classifier

Definition: A Voting Classifier[14] is an ensemble learning technique that combines the predictions of multiple base classifiers and aggregates them to make a final prediction. It can be implemented using different voting strategies such as majority voting or weighted voting.

Usage in Project: In the project, a Voting Classifier [14] is employed to combine the predictions of individual machine learning models such as Random Forest and AdaBoost. By leveraging the collective wisdom of multiple classifiers, the Voting Classifier enhances the accuracy and robustness of network intrusion detection.

Stacking Classifier

Definition: A Stacking Classifier [15] is an ensemble learning method that combines the predictions of multiple base classifiers using another classifier, known as a meta-learner. The meta-learner learns how to best combine the base classifiers' predictions to make the final decision.

Usage in Project: In the project, a Stacking Classifier[15] is utilized to aggregate the predictions of base classifiers such as Random Forest, Decision Tree with LightGBM, and others. By leveraging the diverse expertise of different classifiers, the Stacking Classifier enhances the overall performance of network intrusion detection.

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

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

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)

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

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.

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

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

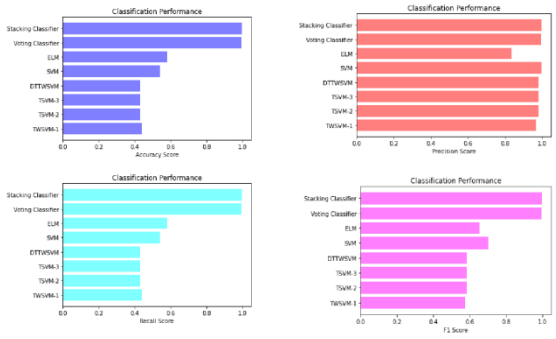

Fig 4 COMPARISON GRAPHS OF NSL-KDD DATASET

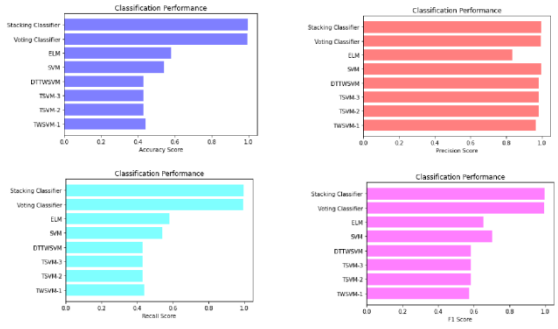

Fig 5 COMPARISON GRAPHS OF UNSW-NB15 DATASET

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

ML Model	Accuracy	Precision	Recall	F1-Score
TWSVM-1	0.440	0.966	0.440	0.574
TSVM-2	0.432	0.983	0.432	0.583
TSVM-3	0.432	0.983	0.432	0.583
DTTWSVM	0.432	0.983	0.432	0.583
SVM	0.541	0.998	0.541	0.702
ELM	0.580	0.834	0.580	0.654
Extension Voting Classifier	0.996	0.996	0.996	0.996
Extension Stacking Classifier	0.999	0.999	0.999	0.999

Fig 6 PERFORMANCE EVALUATION- NSL KDD DATASET

ML Model	Accuracy	Precision	Recall	F1-Score
TWSVM-1	0.440	0.966	0.440	0.574
TSVM-2	0.432	0.983	0.432	0.583
TSVM-3	0.432	0.983	0.432	0.583
DTTWSVM	0.432	0.983	0.432	0.583
SVM	0.541	0.998	0.541	0.702
ELM	0.580	0.834	0.580	0.654
Voting Classifier	0.996	0.996	0.996	0.996
Stacking Classifier	0.999	0.999	0.999	0.999

Fig 7 PERFORMANCE EVALUATION-UNSW NB15 DATASET

Fig 8 home page

Fig 9 sign up

Fig 10 sign in

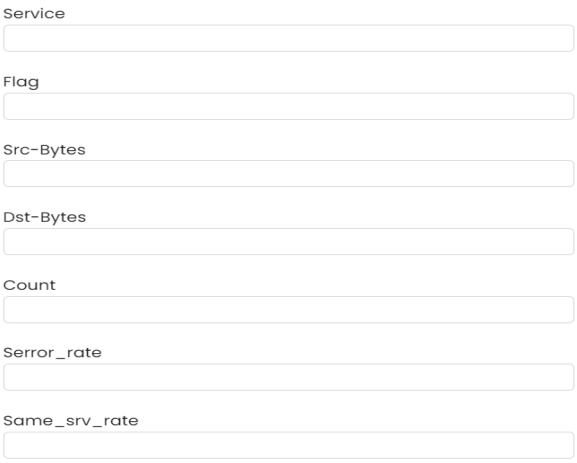

Fig 11 upload input data

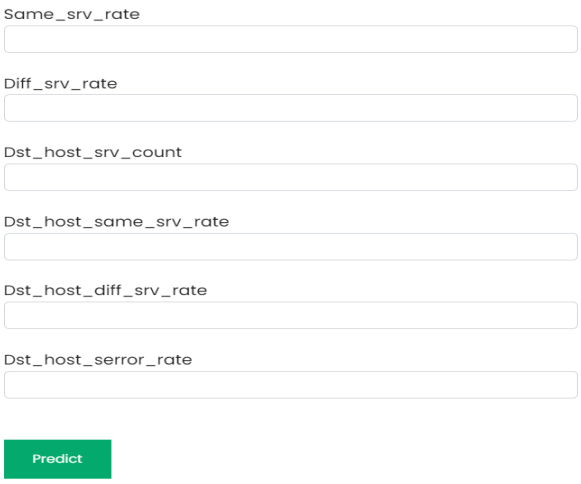

Fig 12 upload input data

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

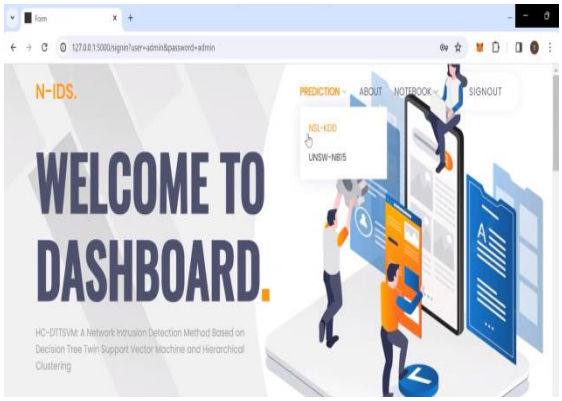

Fig 13 dash board

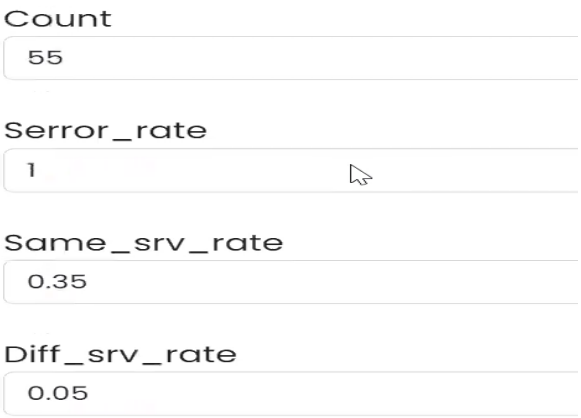

Fig 14 upload input data

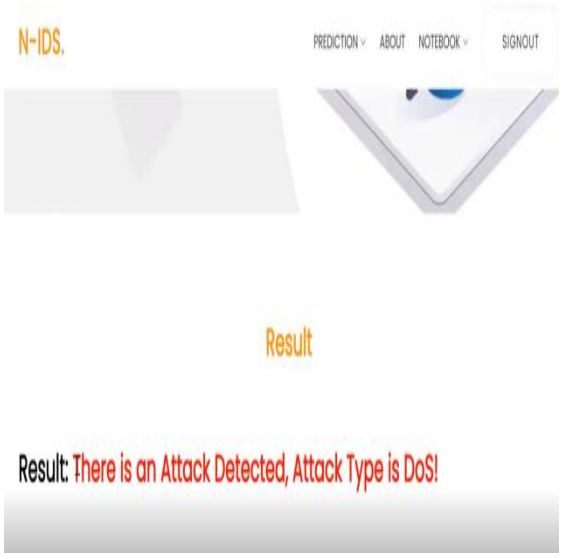

Fig 15 predicted result

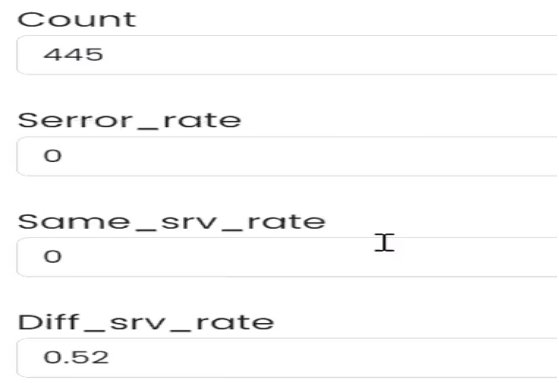

Fig 16 upload input data

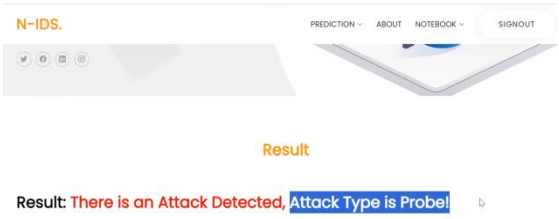

Fig 17 predicted result

Similarly we can try another inputs data to predict results for given input data

5. CONCLUSION

In conclusion, the proposed DTTWSVM method demonstrates promising capabilities in network intrusion detection by effectively leveraging decision tree TWSVM models. Through the iterative training process on subsets of the dataset, DTTWSVM achieves comparable detection performance to recent methods while effectively addressing different categories of network intrusions. Furthermore, the integration of ensemble techniques like Voting Classifier and Stacking Classifier enhances the system's accuracy and robustness. Additionally, the development of a user-friendly Flask interface with secure authentication enhances the overall user experience during system testing, facilitating ease of use and

ensuring data security. Moving forward, the combination of innovative detection methods with user-centric design approaches promises to further strengthen network security and mitigate the risks posed by evolving cyber threats.

6. FUTURE SCOPE

The feature scope of HC-DTTSVM encompasses the integration of decision tree twin support vector machine (DTTSVM) with hierarchical clustering for network intrusion detection. This hybrid approach aims to leverage the strengths of both techniques to enhance the detection accuracy and efficiency of the intrusion detection system. By employing hierarchical clustering, the dataset is partitioned into clusters based on similarities in network traffic patterns, enabling the creation of a hierarchical structure. Subsequently, DTTSVM is applied to each cluster to construct decision trees tailored to specific intrusion patterns within each cluster. This facilitates a more granular and targeted approach to intrusion detection, allowing for nuanced detection of diverse network intrusion scenarios. The feature scope encompasses the implementation and optimization of the hierarchical clustering algorithm, as well as the integration of DTTSVM for decision tree construction within each cluster, ultimately contributing to a robust and adaptive network intrusion detection methodology.

REFERENCES

[1] M. Humayun, M. Niazi, N. Jhanjhi, M. Alshayeb, and S. Mahmood, ‘‘Cyber security threats and vulnerabilities: A systematic mapping study,’’ Arabian J. Sci. Eng., vol. 45, no. 4, pp. 3171–3189, Apr. 2020.

[2] J.-X. Wu, J.-H.Li, and X.-S. Ji, ‘‘Security for cyberspace: Challenges and opportunities,’’ Frontiers Inf. Technol. Electron. Eng., vol. 19, no. 12, pp. 1459–1461, Dec. 2018.

[3] S. Wang, ‘‘Research on the method of network space security countermea sure drill,’’ Int. J. Commun.Syst., vol. 35, no. 5, p. e4654, Mar. 2022.

[4]P.L.GoethalsandM.E.Hunt,‘‘Areviewofscientific researchindefensive cyberspace operation tools and technologies,’’ J. Cyber Secur. Technol., vol. 3, no. 1, pp. 1–46, Jan. 2019.

[5] T. M. Cheung, ‘‘The rise of China as a cybersecurity industrial power: Balancing national security, geopolitical, and development priorities,’’ J. Cyber Policy, vol. 3, no. 3, pp. 306–326, Sep. 2018.

[6]Z.Yang,X.Liu,T.Li,D.Wu,J.Wang,Y.Zhao,andH. Han,‘‘Asystematic literature review of methods and datasets for anomaly-based network intrusion detection,’’ Comput. Secur., vol. 116, May 2022, Art. no. 102675.

[7] M. Mehmood, T. Javed, J. Nebhen, S. Abbas, R. Abid, G. R. Bojja, and M. Rizwan, ‘‘A hybrid approach for network intrusion detection,’’ Comput., Materials Continua, vol. 70, no. 1, pp. 91–107, 2022.

[8] A. Thakkar and R. Lohiya, ‘‘A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions,’’ Artif. Intell. Rev., vol. 55, no. 1, pp. 453–563, Jan. 2022.

[9] L. Yi, M. Yin, and M. Darbandi, ‘‘A deep and systematic review of the intrusion detection

systems in the fog environment,'' Trans. Emerg.Telecommun.Technol., vol. 34, no. 1, p. e4632, Jan. 2023.

[10] R. Khemchandani and S. Chandra, ''Twin support vector machines for pattern classification,'' IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 5, pp. 905–910, May 2007.

[11] S. Ding, N. Zhang, X. Zhang, and F. Wu, ''Twin support vector machine: Theory, algorithm and applications,'' Neural Comput. Appl., vol. 28, no. 11, pp. 3119–3130, Nov. 2017.

[12] S. Ding, J. Yu, B. Qi, and H. Huang, ''An overview on twin support vector machines,'' Artif. Intell. Rev., vol. 42, no. 2, pp. 245–252, Aug. 2014.

[13] H. Huajuan, W. Xiuxi, and Z. Yongquan, ''Twin support vector machines: Asurvey,'' Neurocomputing, vol. 300, pp. 34–43, Jul. 2018.

[14] D. Tomar and S. Agarwal, ''A comparison on multi-class classification methods based on least squares twin support vector machine,'' Knowl. Based Syst., vol. 81, pp. 131–147, Jun. 2015.

[15]S.Ding,X.Zhao,J.Zhang,X.Zhang,andY.Xue,'' Areviewonmulti-class TWSVM,'' Artif. Intell. Rev., vol. 52, no. 2, pp. 775–801, Aug. 2019.

[16] A. A. Aburomman and M. B. I. Reaz, ''A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems,'' Inf. Sci., vol. 414, pp. 225–246, Nov. 2017.

[17] M.Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, ''A detailed analysis of the KDD CUP99data set,'' in Proc. IEEE Symp. Comput.Intell.Secur.Defense Appl., Jul. 2009, pp. 1–6.

[18] N. Moustafa and J. Slay, ''UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),'' in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Nov. 2015, pp. 1–6.

[19] H. Liu and B. Lang, ''Machine learning and deep learning methods for intrusion detection systems: A survey,'' Appl. Sci., vol. 9, no. 20, p. 4396, Oct. 2019.

[20] R.Magán-Carrión, D. Urda, I. Díaz-Cano, and B. Dorronsoro, ''Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches,'' Appl. Sci., vol. 10, no. 5, p. 1775, Mar. 2020.

[21] Q.-V. Dang, ''Using machine learning for intrusion detection systems,'' Comput. Informat., vol. 41, no. 1, pp. 12–33, 2022.

[22] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, ''Machine learning and deep learning approaches for cybersecuriy: A review,'' IEEE Access, vol. 10, pp. 19572–19585, 2022.

[23] A. Das, S. A. Ajila, and C. H. Lung, ''A comprehensive analysis of accuracies of machine learning algorithms for network intrusion detection,'' in Proc. Int. Conf. Mach. Learn.Netw. Cham, Switzerland: Springer, 2019, pp. 40–57.

[24] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. A. Khan, ''Performance analysis of machine learning algorithms in intrusion detection system: A review,'' Proc. Comput. Sci., vol. 171, pp. 1251–1260, Jan. 2020.

[25] B. Kumar, O. P. Vyas, and R. Vyas, ‘‘A comprehensive review on the variants of support vector machines,’’ Modern Phys. Lett.B, vol. 33, no. 25, Sep. 2019, Art.no. 1950303.

[26] J. Liu, J. Feng, and X. Gao, ‘‘Fault diagnosis of rod pumping wells based on support vector machine optimized by improved chicken swarm optimization,’’ IEEE Access, vol. 7, pp. 171598–171608, 2019.

[27] L. C. Padierna, C. Villasenor-Mora, and S. A. Lopez Juarez, ‘‘Biomedical classification problems automatically solved bycomputationalintelligence methods,’’ IEEE Access, vol. 8, pp. 101104–101117, 2020.

[28]S.Teng,N.Wu,H.Zhu,L.Teng,andW.Zhang,‘‘SVM-DT-basedadaptive and collaborative intrusion detection,’’ IEEE/CAA J. Autom. Sin., vol. 5, no. 1, pp. 108–118, Jan. 2018.

[29] J. Gu and S. Lu, ‘‘An effective intrusion detection approach using SVM with naïveBayesfeatureembedding,’’Comput.Secur.,vol. 103,Apr.2021, Art. no. 102158.

[30] D. Jing and H.-B. Chen, ‘‘SVM based network intrusion detection for the UNSW-NB15 dataset,’’ in Proc. IEEE 13th Int. Conf. ASIC (ASICON), Oct. 2019, pp. 1–4.