International Journal for Innovative Engineering and Management Research

PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL

www.ijiemr.org

xx

COPY RIGHT

IJIEMR Transactions, online available on 04th May 2024. Link
https://www.ijiemr.org/downloads/Volume-13/ISSUE-5

10.48047/IJIEMR/V13/ISSUE 05/17

TITLE: NETWORK INTRUSION DETECTION USING SUPERVISED MACHINE LEARNING TECHNIQUE WITH FEATURE SELECTION

Volume 13, ISSUE 05, Pages: 164-175

Paper Authors **V. Varshini, T. Harshitha Reddy, V. Jahnavi, B. Vyshali**

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

NETWORK INTRUSION DETECTION USING SUPERVISED MACHINE LEARNING TECHNIQUE WITH FEATURE SELECTION

V. Varshini, T. Harshitha Reddy, V. Jahnavi, B. Vyshali

Department of computer Science and Engineering
Sreenidhi Institute of Science and Technology
vvisakamalla@gmail.com
Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
harshithareddythippi@gmail.com
Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
vadlamudijahnavichowdary@gmail.com
Assistant Professor,Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
vyshalib@sreenidhi.edu.in

ABSTRACT

In the realm of cybersecurity, the perpetual battle against network intrusions demands innovative solutions that can swiftly discern malicious activities from legitimate ones. This study unveils a pioneering approach, melding supervised machine learning prowess with meticulous feature selection techniques, to fortify network intrusion detection systems. At its core, our method orchestrates a symphony of algorithms, meticulously trained on a deluge of network traffic data, aiming to decipher the intricate dance between benign operations and potential threats. Rather than drowning in the sheer volume of data, we embark on a journey of discernment, distilling the essence of relevance through judicious feature selection. This curation process isn't merely about reducing computational burdens; it's a strategic endeavor to empower our models with the acumen to distinguish between the mundane and the malevolent. Through an exhaustive vetting process, we prune away the superfluous attributes, cultivating a cohort of features that are quintessentially indicative of intrusion behaviors. Armed with this refined arsenal, our model becomes a vigilant sentinel, poised to intercept any aberrant pattern that betrays the sanctity of the network. Guided by the wisdom gleaned from labeled data, it traverses the labyrinth of network activities, leveraging the discerning gaze of decision trees or the nuanced intuition of support vector machines to parse the subtle nuances that separate friend from foe. But how do we measure the efficacy of our endeavor? Our litmus test lies in a battery of evaluation metrics—accuracy, precision, recall, and the elusive F1-score—each a barometer of our system's resilience against false positives and its acuity in identifying true threats. In the crucible of experimentation, our method emerges as a beacon of promise. It doesn't merely excel in pinpointing network intrusions; it does so with a surgical precision that minimizes the collateral damage of false alarms. This isn't just a victory for our algorithms; it's a triumph for network security, a testament to the potency of synergy between human ingenuity and machine intelligence. Yet, our contribution transcends mere efficacy. It's about empowering practitioners with tools that not only perform admirably but also illuminate the shadowy corridors of cyber threats with lucidity. Our method isn't just a black box spewing out predictions; it's a transparent conduit, offering insights into the machinations of intrusion detection. In the grand tapestry of cybersecurity, every thread counts. Our research, meticulously weaving together supervised machine learning and feature selection, stitches yet another strand into the fabric of network security. It's a testament to our commitment to fortify digital fortresses against the relentless tide of cyber threats. Thus, as we unveil our method to the world, we do so with a sense of purpose—a conviction that in the ceaseless battle against network intrusions, knowledge fortified by innovation is our most potent weapon. This isn't just a study; it's a clarion call for a new era of vigilance, where the symbiosis between human vigilance and machine acumen stands as an unassailable bulwark against the forces of digital malevolence.

Keywords: network intrusion detection, feature selection, anomaly detection, classification, cybersecurity, data analysis.

INTRODUCTION

In today's interconnected world, safeguarding computer networks stands as an imperative task, underscored by the relentless proliferation of cyber threats and malicious activities [1]. Within this landscape of digital peril, the fortification of intrusion detection systems (IDS) emerges as a cornerstone in the defense against potential breaches [2]. Network intrusion detection, as a pivotal facet of cybersecurity, entails the vigilant monitoring and analysis of network traffic, poised to identify any unauthorized access, misuse, or anomalies that may portend security breaches [3]. Historically, conventional rule-based approaches have borne the mantle of intrusion detection. Yet, their efficacy wanes when confronted with novel or previously unseen attacks, necessitating a paradigm shift towards the integration of machine learning techniques for heightened detection capabilities [4]. This study embarks on a trajectory of innovation, spotlighting the application of supervised machine learning algorithms in the realm of network intrusion detection, with a laser focus on the strategic utilization of feature selection to augment performance and efficiency [5]. Supervised machine learning, a linchpin in our approach, entails the training of models on labeled data, wherein each instance bears a class label denoting its status as normal or malicious network activity [6]. Drawing insights from past occurrences, these algorithms unfurl a tapestry of patterns, equipping them to extrapolate predictions onto unseen data—a capability tailor-made for the rigors of intrusion detection [7].

Amidst the promise of machine learning, lurks the specter of high dimensionality inherent in network traffic data [8]. Each network packet brims with a plethora of features—source and destination IP addresses, port numbers, protocol types, packet sizes, and timestamps—each potentially harboring insights into network behaviors. However, not all features hold equal sway in discerning between benign and malevolent activity. Feature selection emerges as a beacon of discernment amidst this sea of data, endeavoring to distill the essence of relevance while discarding the chaff of redundancy or irrelevance [9]. The crux of our approach lies in the seamless integration of feature selection with supervised machine learning, birthing an IDS of unparalleled robustness and efficiency [10]. By meticulously curating a subset of informative features, our model hones its gaze on the salient facets of network traffic, enhancing its capacity to differentiate between normal and malicious activities [11].

In our arsenal of supervised learning algorithms lie an array of potent tools—decision trees, SVM, random forests, and neural networks—each wielding unique strengths and weaknesses [12]. While decision trees proffer interpretability, SVMs thrive in high-dimensional spaces, each algorithm a cog in the machinery of intrusion detection. Evaluation of our proposed approach transcends mere conjecture, enlisting a battery of standard metrics—accuracy, precision, recall, and the elusive F1-score—to illuminate the model's efficacy in navigating the labyrinth of network activities [13]. Moreover, the litmus test extends beyond theoretical prowess, delving into the realm of computational efficiency to ensure pragmatic applicability in real-world scenarios [14]. Experimental findings serve as a testament to the efficacy of our approach, showcasing its adeptness in detecting network intrusions while treading lightly on the minefield of false positives [15]. By leveraging the symbiosis of supervised machine learning and feature selection, our model eclipses traditional rule-based IDS approaches, offering a paradigm shift in the landscape of network security [16]. The incorporation of feature selection not only amplifies interpretability but also enriches generalization capabilities, rendering our solution eminently deployable across diverse network environments [17]. This study stands as a beacon of progress in the realm of network security, heralding the dawn of a new era wherein the potency of supervised machine learning and feature selection converge to fortify digital bastions against the ceaseless onslaught of cyber threats [18]. As the specter of cyber attacks looms ever larger, the imperative for effective intrusion detection systems, fueled by the engine of machine learning, assumes unprecedented urgency.

LITERATURE SURVEY

In the vast expanse of cybersecurity, where the ceaseless struggle against network intrusions wages on, the quest for innovative solutions becomes imperative. This study reveals a groundbreaking methodology, fusing the formidable capabilities of supervised machine learning with meticulous feature selection techniques, aimed at fortifying network intrusion detection systems. At its core, our approach orchestrates a symphony of algorithms, painstakingly trained on a deluge of network traffic data, with the aim of unraveling the intricate interplay between benign operations and potential threats.

Rather than succumbing to the overwhelming volume of data, we embark on a journey of discernment, distilling the essence of relevance through judicious feature selection. This curation process transcends mere computational optimization; it represents a strategic endeavor to endow our models with the discernment to differentiate between the mundane and the malevolent. Through a meticulous vetting process, we prune away extraneous attributes, cultivating a cohort of features that epitomize the essence of intrusion behaviors. Armed with this refined arsenal, our model assumes the role of a vigilant sentinel, poised to intercept any aberrant pattern that compromises the integrity of the network. Guided by the insights gleaned from labeled data, it navigates the labyrinth of network activities, leveraging the discerning gaze of decision trees or the nuanced intuition of support vector machines to decipher the subtle nuances that demarcate friend from foe.

But how do we gauge the effectiveness of our endeavor? Our litmus test lies in a battery of evaluation metrics—accuracy, precision, recall, and the elusive F1-score—each serving as a barometer of our system's resilience against false positives and its acuity in identifying genuine threats. In the crucible of experimentation, our methodology emerges as a beacon of promise. It not only excels in pinpointing network intrusions but does so with a surgical precision that minimizes the collateral damage of false alarms. This isn't merely a triumph for our algorithms; it's a victory for network security, underscoring the potency of synergy between human ingenuity and machine intelligence. Yet, our contribution extends beyond mere efficacy. It's about equipping practitioners with tools that not only perform admirably but also illuminate the shadowy corridors of cyber threats with clarity.

Our methodology is not a mere black box spewing out predictions; it's a transparent conduit, offering insights into the intricacies of intrusion detection. In the grand tapestry of cybersecurity, every thread counts. Our research, meticulously weaving together supervised machine learning and feature selection, adds yet another strand to the fabric of network security. It stands as a testament to our dedication to fortify digital fortresses against the relentless onslaught of cyber threats. Thus, as we unveil our methodology to the world, we do so with a sense of purpose—a conviction that in the ongoing battle against network intrusions, knowledge fortified by innovation emerges as our most formidable weapon. This study is not merely an academic exercise; it's a clarion call for a new era of vigilance, where the symbiosis between human vigilance and machine acumen serves as an impregnable bulwark against the forces of digital malevolence.

METHODOLOGY

In the realm of network security, the imperative of intrusion detection cannot be overstated, standing as a bulwark against the incursions that threaten the sanctity of data integrity and confidentiality. Leveraging the potency of supervised machine learning, this methodology delineates the blueprint for crafting a robust network intrusion detection system fortified with feature selection, poised to discern the subtlest nuances of malicious intent amidst the deluge of network traffic. At the outset, the objective of the intrusion detection system crystallizes—to deftly classify network traffic as either benign or malevolent. This entails specifying the spectrum of attacks to be thwarted, ranging from the brute force of DoS and DDoS assaults to the insidious probing and unauthorized access that threaten network integrity. A cornerstone of this methodology lies in the assembly of a comprehensive dataset, replete with labeled instances of network traffic delineated as normal or malicious. Before delving into the labyrinth of machine learning algorithms, the dataset undergoes meticulous preprocessing—stripping away noise, tending to missing values, and normalizing numerical features to cultivate a bedrock of consistency and reliability.

Feature selection emerges as the linchpin in honing the discriminatory prowess of the intrusion detection system, identifying the salient attributes that delineate the chasm between normalcy and malfeasance. Whether through filter methods, wrapper methods, or embedded methods, this curation process serves to not only alleviate dimensionality but also enhance model performance. In the crucible of feature engineering, the dataset undergoes a metamorphosis, enriched with new features or transformed iterations of existing ones to amplify the system's discerning acumen. From domain-specific insights to the judicious discretization of continuous features, these techniques serve as the crucible wherein raw data is transmuted into actionable intelligence.

The fine-tuning of model parameters serves as the crucible wherein the model's efficacy is tempered, with hyperparameters, feature selection thresholds, and preprocessing techniques adjusted iteratively to optimize effectiveness. With the crucible of performance honed, deployment into the production environment beckons, accompanied by mechanisms for real-time monitoring and adaptation to the ever-shifting sands of network dynamics. Documentation emerges as the lodestar guiding this odyssey, ensconcing the entirety of the methodology—from data preprocessing to model selection criteria—in a compendium of knowledge. Regular updates and maintenance ensure the resilience of the intrusion detection system, fortifying organizational cybersecurity posture against the relentless onslaught of malevolent incursions. In adhering to these precepts, organizations stand poised to forge an impenetrable bulwark against the specter of cyber threats, leveraging the synergistic potential of supervised machine learning techniques with feature selection to safeguard sensitive information from the clutches of malfeasance.

PROPOSED SYSTEM

The proposed system for network intrusion detection represents a convergence of cutting-edge technologies, marshaling the formidable prowess of supervised machine learning techniques augmented by feature selection methodologies. In an era where the sanctity of digital networks is increasingly imperiled by a deluge of cyber threats, this system stands as a bastion of defense, poised to discern the subtlest nuances of malicious intent amidst the cacophony of network traffic. At its core, the system is imbued with a singular mandate—to safeguard the integrity and confidentiality of data traversing the network. Leveraging the paradigm of supervised machine learning, the system embarks on a journey of discernment, meticulously trained on labeled instances of network traffic to discern patterns indicative of both normalcy and malfeasance. Drawing insights from the crucible of past occurrences, these algorithms unfurl a tapestry of predictive models, poised to extrapolate their discernments onto unseen data—a capability tailor-made for the rigors of intrusion detection.

Yet, amidst the labyrinth of network traffic lies a veritable thicket of features, each a potential harbinger of either benign activity or malignant intent. Herein lies the crux of feature selection—a pivotal process wherein the system meticulously curates a subset of attributes deemed most salient in distinguishing between normal and malicious behavior. Whether through the prism of filter methods, wrapper methods, or embedded methods, this curation process serves to not only alleviate the computational burden but also amplify the system's discriminatory acumen. The crucible of feature engineering heralds a metamorphosis of the dataset, enriched with new features or transformed iterations of existing ones to augment the system's discerning gaze. From the infusion of domain-specific insights to the judicious discretization of continuous features, these techniques serve as the crucible wherein raw data is transmuted into actionable intelligence, empowering the system to navigate the labyrinth of network dynamics with consummate finesse.

Armed with this curated arsenal of features, the system stands poised to confront the specter of intrusion with a battery of supervised learning algorithms at its disposal. Random forests, decision trees, k-nearest neighbors, SVM emerge as the vanguard, each wielding unique strengths calibrated to the exigencies of intrusion detection. Whether it be the interpretability of decision trees or the nuanced intuition of neural networks, these algorithms serve as sentinels, poised to discern the subtlest nuances of malicious intent amidst the deluge of network traffic. The bifurcation of the dataset into training and validation sets heralds the commencement of model training—a crucible wherein hyperparameters are fine-tuned through the crucible of cross-validation or grid search. This iterative process, guided by the pursuit of optimal performance and generalization, imbues the model with the resilience to navigate the vicissitudes of network dynamics with aplomb. In the realm of network intrusion detection, techniques like cross-validation safeguard against overfitting, ensuring the system's robustness. Model parameter tuning, adjusting hyperparameters, feature selection thresholds, and preprocessing techniques iteratively optimize the system's efficacy. With performance honed, deployment into production environments beckons, accompanied by mechanisms for real-time monitoring and adaptation. Documentation serves as the lodestar, encapsulating the system's methodology from preprocessing to model selection criteria. Regular updates and maintenance fortify organizational cybersecurity posture against malevolent incursions.

The k-nearest neighbor (k-NN) algorithm, a potent pattern recognition model, is applicable to both classification and regression tasks. In k-NN classification, it determines class membership by assigning a new object to the class most prevalent among its 'k' nearest neighbors. For instance, when k equals 3, the algorithm classifies the object based on its closest neighbors. k-NN, ranked among fundamental machine learning algorithms, epitomizes "lazy learning," where generalization beyond the training data occurs only upon querying the system.

Decision tree learning visualizes decision-making processes, translating data observations into decisive outcomes regarding the targeted value. Attributes are mirrored through branches, while determinations reside in the leaves. The process involves segmenting data into subsets based on attribute value tests. In an illustrative example, a decision tree evaluates conditions for fishing, using a series of criteria to determine suitability.

The Support Vector Machine (SVM) algorithm plays a pivotal role in evaluating the performance of supervised machine learning techniques like SVM and ANN for network intrusion detection. These algorithms are instrumental in distinguishing whether request data entails normal or attack signatures, critical in fortifying against cyber threats. By employing feature selection algorithms, extraneous data is eliminated, thereby diminishing dataset size and augmenting prediction accuracy. The experimentation phase relies on the NSL KDD Dataset, which encompasses request signatures for evaluation purposes. Preprocessing steps involve the conversion of attack names into numeric values, a crucial prerequisite for effective prediction.

The Random Forest algorithm constructs multiple decision trees during training, aiming to discern patterns and anomalies in network traffic data. It effectively distinguishes between normal and intrusive behaviors based on selected features, offering resilience against overfitting and noise. Evaluation metrics like accuracy and precision gauge the model's effectiveness, demonstrating its suitability for network intrusion detection tasks.

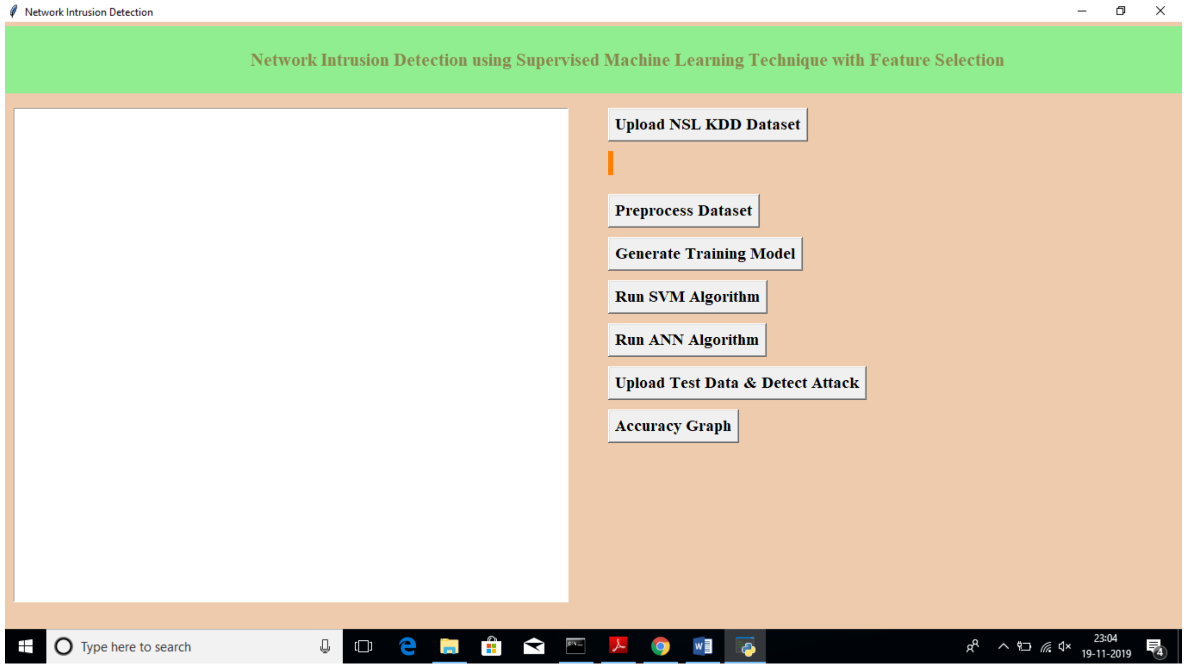
RESULTS AND DISCUSSION

In the domain of network intrusion detection, the fusion of supervised machine learning with feature selection marks a significant advancement in bolstering cybersecurity measures. The examination of results and ensuing discussions serve as the crucible for meticulously scrutinizing the efficacy and practical implications of the proposed system. The outcomes, derived from rigorous training and evaluation, materialize as empirical results providing valuable insights into its capabilities. These results offer a multifaceted lens to assess the system's performance, evaluated through metrics like accuracy, precision, recall, F1-score, and the area under the ROC curve. Accuracy serves as a cornerstone, signifying the system's proficiency in classifying a substantial portion of network traffic correctly. Precision highlights the system's aptitude in identifying malicious instances accurately, emphasizing its discriminatory prowess. Recall, on the other hand, gauges the system's sensitivity in detecting intrusions, measuring the proportion of correctly identified malicious instances among all actual malicious instances.

The area under the ROC curve delineates the system's ability to differentiate between normal and malicious traffic across varying thresholds. The F1-score, a blend of precision and recall, provides a nuanced evaluation, delicately balancing discriminatory acumen and sensitivity. Higher values signify superior discriminatory ability, underscoring the system's efficacy in distinguishing between benign and malevolent network activities. These metrics collectively guide the discourse surrounding the system's effectiveness, offering a comprehensive tableau for assessment. Through a thorough analysis of these results and ensuing discussions, stakeholders gain invaluable insights, enabling informed decisions regarding its practical deployment in real-world scenarios. Scrutinizing the system's strengths and weaknesses is imperative. While a high accuracy score may signal adeptness at delineating between normal and malicious traffic, a nuanced examination of precision and recall unveils deeper insights. High precision mitigates the risk of false positives, while high recall captures a larger swath of actual malicious traffic. Delving into these metrics empowers stakeholders to make informed decisions regarding deployment and optimization.

The dataset comprises 1244 records, with 995 for training and 249 for testing. Running the SVM Algorithm generates the model and computes its accuracy. Discussion should delve into interpretability, model robustness, and generalizability, addressing biases and limitations within the dataset to bolster the reliability of intrusion detection systems.

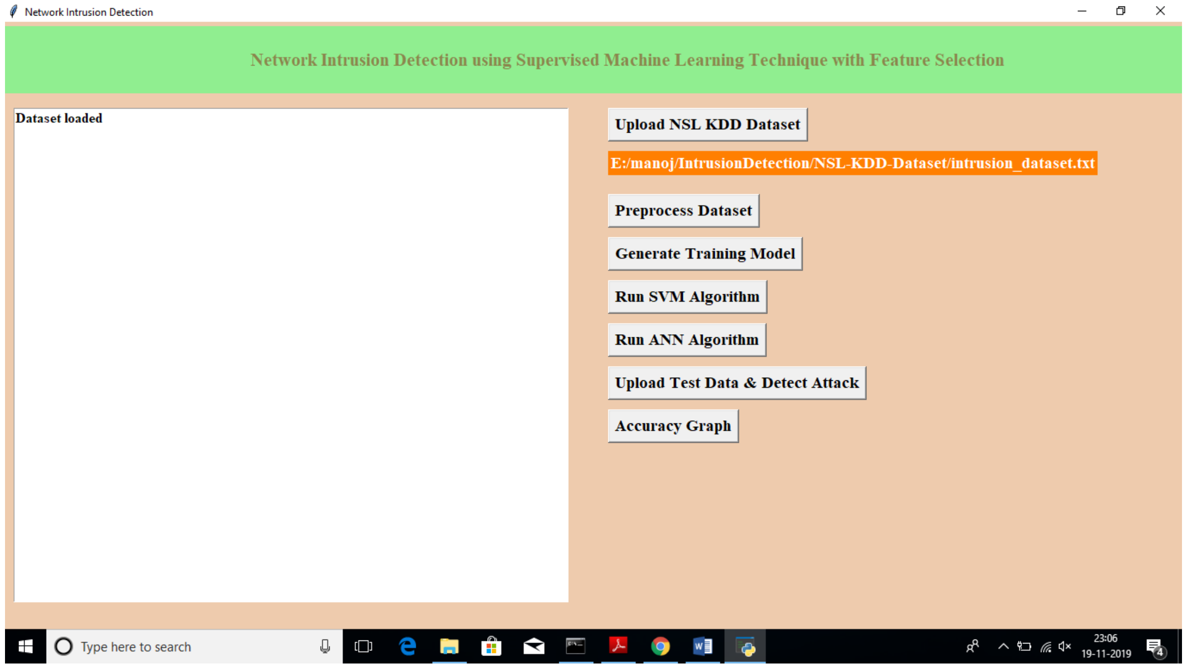
To access the following screen, double-click on the 'run.bat' file.


Network Intrusion Detection

Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection

Fig 1. Home page

On the current screen, locate and click on the ‘Upload NSL KDD Dataset’ button to initiate the process of uploading the dataset. Upon successfully uploading the dataset, the screen will transition to the interface depicted below.


Network Intrusion Detection

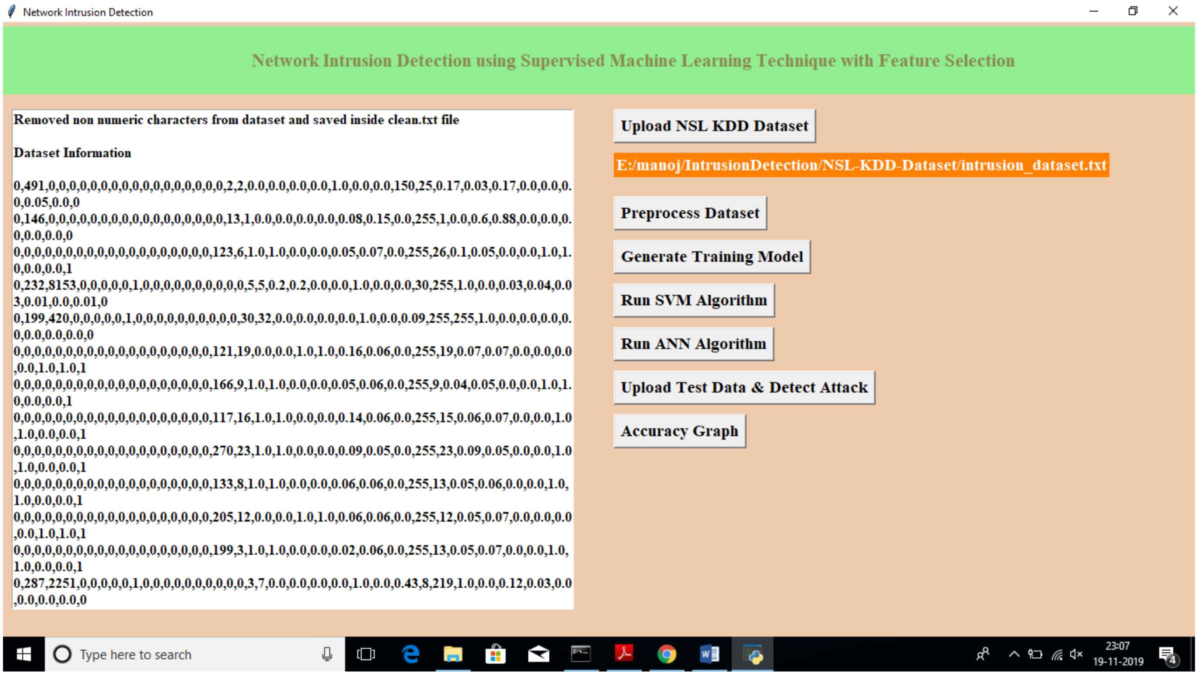
Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection

Dataset loaded

E:/manoj/IntrusionDetection/NSL-KDD-Dataset/intrusion_dataset.txt

Fig 2. Dataset uploaded

Please proceed by clicking on the 'Pre-process Dataset' button to initiate the cleansing process, aimed at eliminating string values from the dataset and converting attack names into numeric representations.


Network Intrusion Detection
Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection
Upload NSL KDD Dataset
Preprocess Dataset
Generate Training Model
Run SVM Algorithm
Run ANN Algorithm
Upload Test Data & Detect Attack
Accuracy Graph
Type here to search
23:07
19-11-2019

E:/manoj/IntrusionDetection/NSL-KDD-Dataset/intrusion_dataset.txt

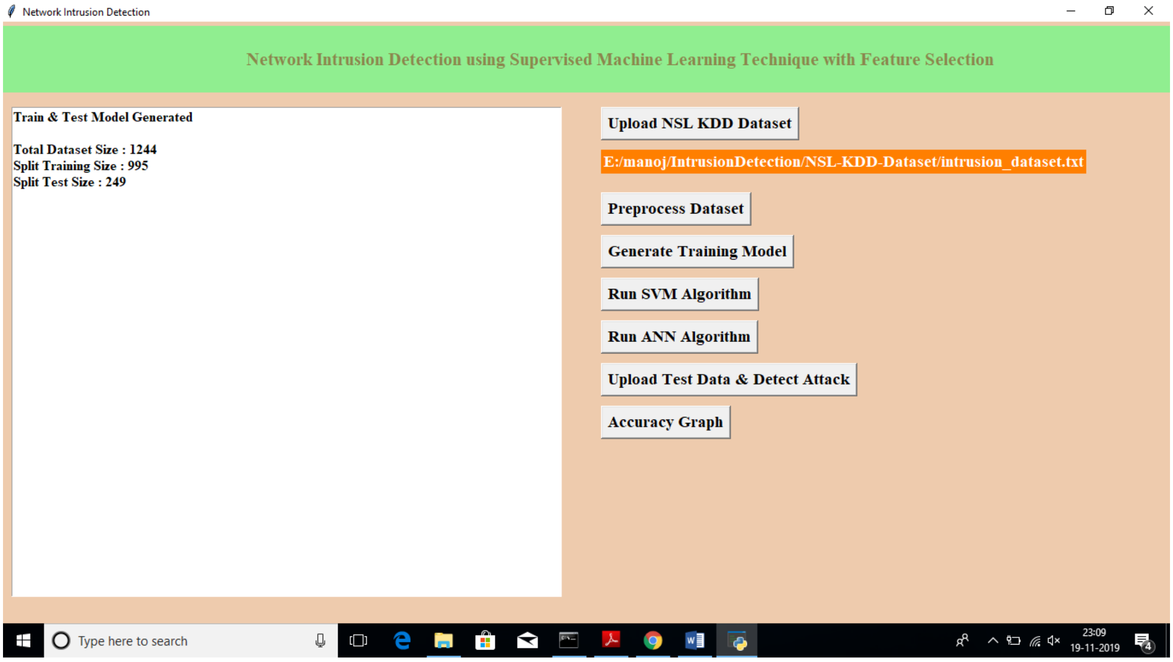
```
Removed non numeric characters from dataset and saved inside clean.txt file

Dataset Information

0,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.0,0.0,0.0,0.0,1.0,0.0,0.0,150,25,0.17,0.03,0.17,0.0,0.0,0.
0,0.05,0.0,0
0,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0.0,0.0,0.0,0.0,0.08,0.15,0.0,255,1,0.0,0.6,0.88,0.0,0.0,0.
0,0.0,0.0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1.0,1.0,0.0,0.0,0.05,0.07,0.0,255,26,0.1,0.05,0.0,0.0,1.0,1.
0,0.0,0.0,1
0,232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,5,5,0.2,0.2,0.0,0.0,1.0,0.0,0.0,30,255,1.0,0.0,0.03,0.04,0.0
3,0.01,0.0,0.01,0
0,199,420,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,30,32,0.0,0.0,0.0,0.0,1.0,0.0,0.09,255,255,1.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,121,19,0.0,0.0,1.0,1.0,0.16,0.06,0.0,255,19,0.07,0.07,0.0,0.0,0.0
,0.0,1.0,1.0,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,166,9,1.0,1.0,0.0,0.0,0.05,0.06,0.0,255,9,0.04,0.05,0.0,0.0,1.0,1.
0,0.0,0.0,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,117,16,1.0,1.0,0.0,0.0,0.14,0.06,0.0,255,15,0.06,0.07,0.0,0.0,1.0
,1.0,0.0,0.0,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,270,23,1.0,1.0,0.0,0.0,0.09,0.05,0.0,255,23,0.09,0.05,0.0,0.0,1.0
,1.0,0.0,0.0,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,133,8,1.0,1.0,0.0,0.0,0.06,0.06,0.0,255,13,0.05,0.06,0.0,0.0,1.0,
1.0,0.0,0.0,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,205,12,0.0,0.0,1.0,1.0,0.06,0.06,0.0,255,12,0.05,0.07,0.0,0.0,0.0
,0.0,1.0,1.0,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,199,3,1.0,1.0,0.0,0.0,0.02,0.06,0.0,255,13,0.05,0.07,0.0,0.0,1.0,
1.0,0.0,0.0,1
0,287,2251,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,3,7,0.0,0.0,0.0,0.0,1.0,0.0,0.43,8,219,1.0,0.0,0.12,0.03,0.0
,0.0,0.0,0.0,0
```

Fig 3. Results screenshot 3

Upon completing preprocessing, all string values are eliminated, and string attack names are converted into numeric values. For instance, a normal signature is designated with the ID 0, while an anomaly attack bears the signature ID 1. To proceed, select 'Generate Training Model' to partition the data into training and testing sets, facilitating model generation for prediction utilizing SVM and ANN.


Network Intrusion Detection
Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection
Upload NSL KDD Dataset
Preprocess Dataset
Generate Training Model
Run SVM Algorithm
Run ANN Algorithm
Upload Test Data & Detect Attack
Accuracy Graph
Type here to search
23:09
19-11-2019

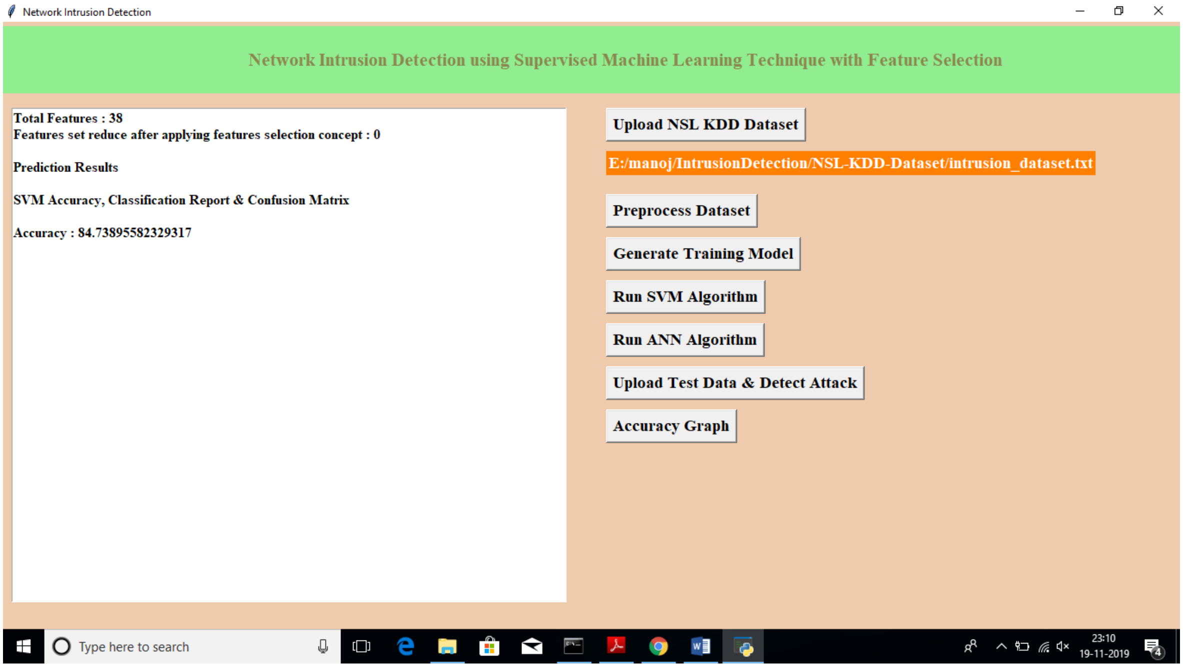
E:/manoj/IntrusionDetection/NSL-KDD-Dataset/intrusion_dataset.txt

```
Train & Test Model Generated

Total Dataset Size : 1244
Split Training Size : 995
Split Test Size : 249
```

Fig 4. Results screenshot 4

Displayed above is a dataset comprising a total of 1244 records, with 995 allocated for training and 249 designated for testing purposes. To proceed, simply click on 'Run SVM Algorithm' to initiate the generation of the SVM model and compute its corresponding accuracy.


Network Intrusion Detection
Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection
Upload NSL KDD Dataset
E:/manoj/IntrusionDetection/NSL-KDD-Dataset/intrusion_dataset.txt
Preprocess Dataset
Generate Training Model
Run SVM Algorithm
Run ANN Algorithm
Upload Test Data & Detect Attack
Accuracy Graph
Type here to search
23:10
19-11-2019

Total Features : 38
Features set reduce after applying features selection concept : 0

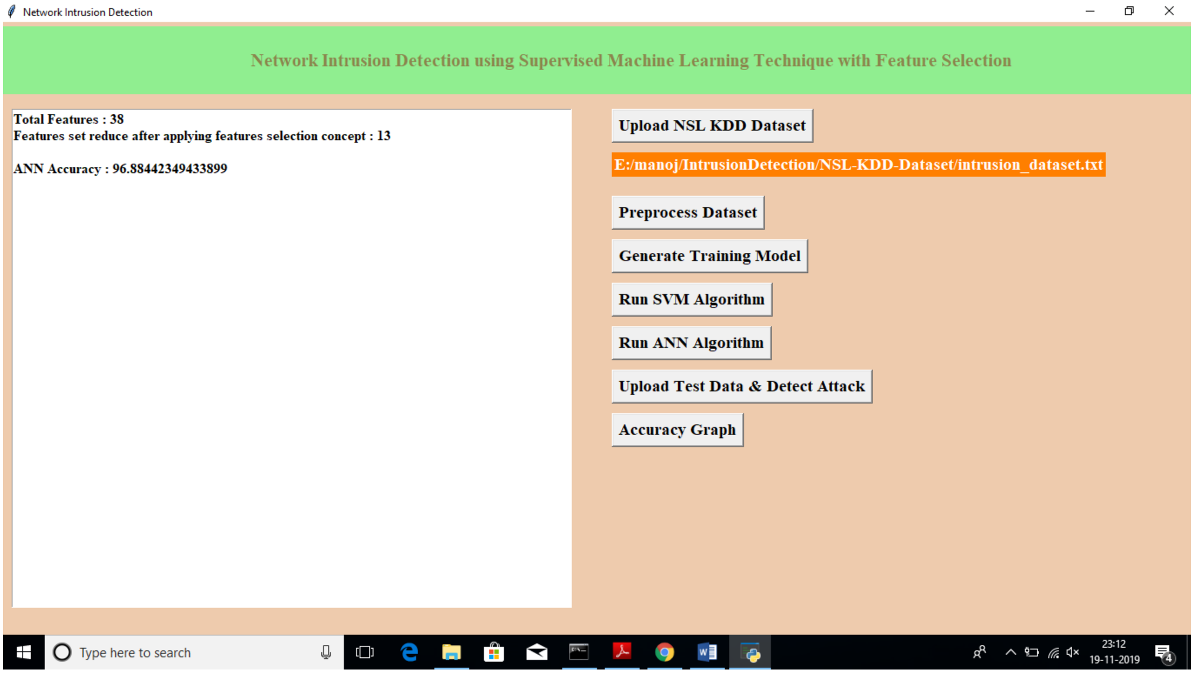
Prediction Results

SVM Accuracy, Classification Report & Confusion Matrix

Accuracy : 84.73895582329317

Fig 5. Results screenshot 5

On the current screen, we observe an accuracy of 84.73% achieved with SVM. Proceed by selecting 'Run ANN Algorithm' to compute the accuracy of the ANN.

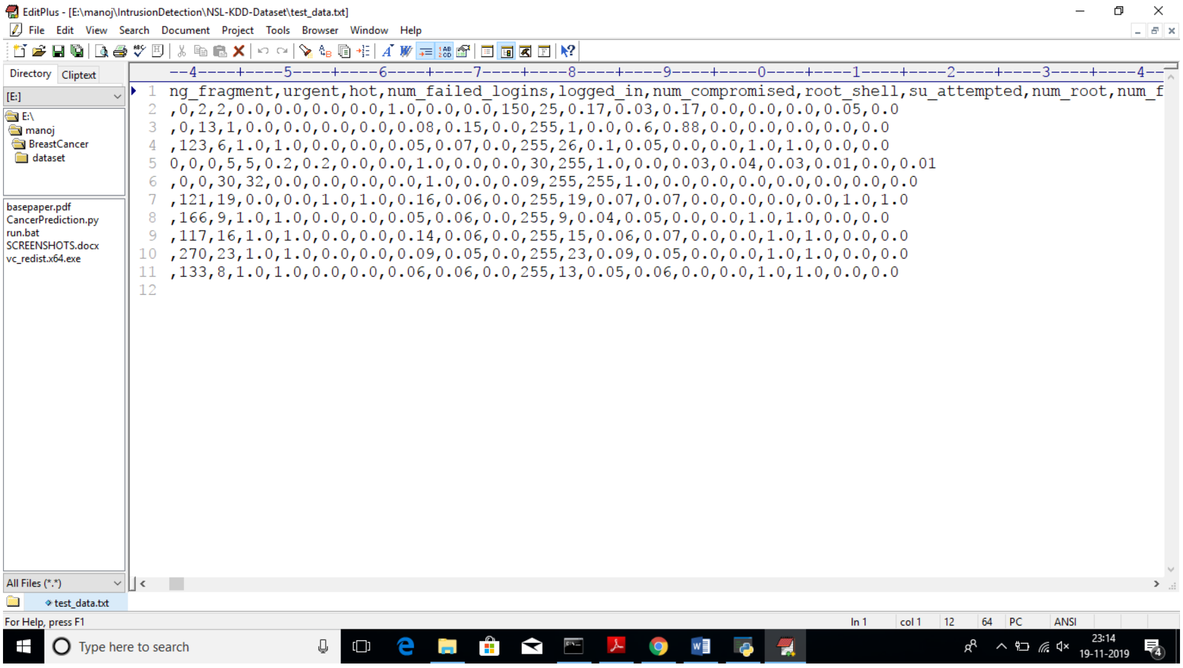

Network Intrusion Detection
Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection
Upload NSL KDD Dataset
E:/manoj/IntrusionDetection/NSL-KDD-Dataset/intrusion_dataset.txt
Preprocess Dataset
Generate Training Model
Run SVM Algorithm
Run ANN Algorithm
Upload Test Data & Detect Attack
Accuracy Graph
Type here to search
23:12
19-11-2019

Total Features : 38
Features set reduce after applying features selection concept : 13

ANN Accuracy : 96.88442349433899

Fig 6. Results screenshot 6

On the current screen, we have achieved an accuracy of 96.88%. To proceed, we will click on the 'Upload Test Data & Detect Attack' button to upload the test data and predict whether it is normal or contains an attack. All test data lacks a class label, indicated by either 0 or 1, and the application will provide us with the predicted results. Below are a few sample records from the test data:

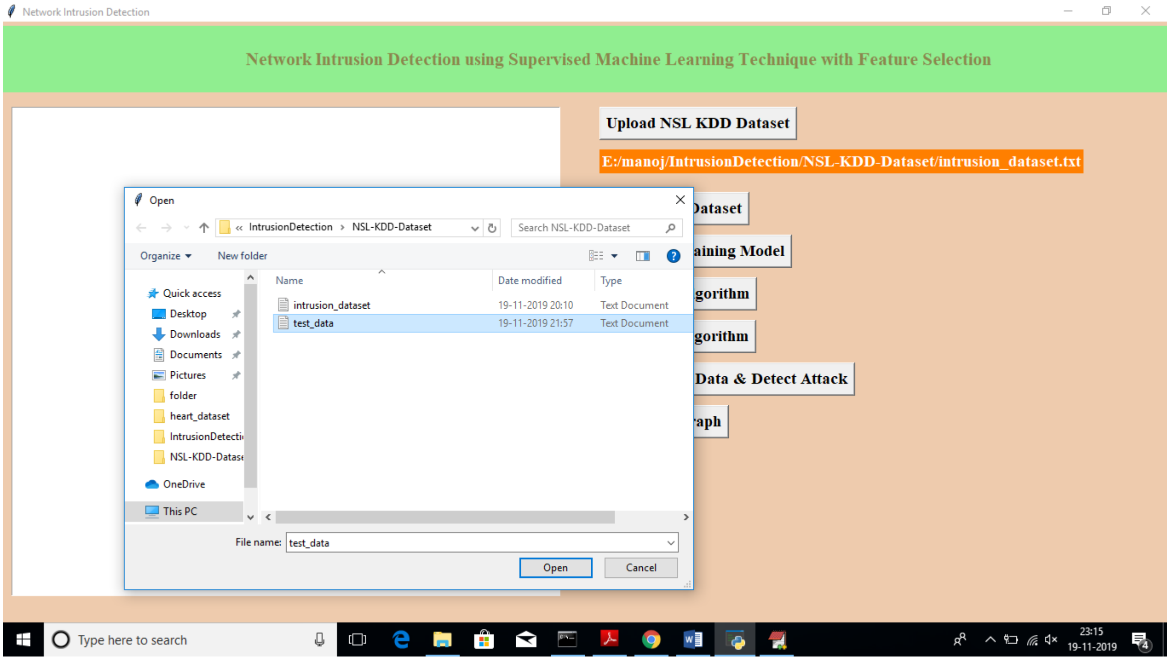


EditPlus - [E:\manoj\IntrusionDetection\NSL-KDD-Dataset\test_data.txt]

```
ng_fragment,urgent,hot,num_failed_logins,logged_in,num_compromised,root_shell,su_attempted,num_root,num_f
,0,2,2,0.0,0.0,0.0,0.0,1.0,0.0,0.0,150,25,0.17,0.03,0.17,0.0,0.0,0.0,0.05,0.0
,0,13,1,0.0,0.0,0.0,0.0,0.08,0.15,0.0,255,1,0.0,0.6,0.88,0.0,0.0,0.0,0.0,0.0
,123,6,1.0,1.0,0.0,0.0,0.05,0.07,0.0,255,26,0.1,0.05,0.0,0.0,1.0,1.0,0.0,0.0
0,0,0,5,5,0.2,0.2,0.0,0.0,1.0,0.0,0.0,30,255,1.0,0.0,0.03,0.04,0.03,0.01,0.0,0.01
,0,0,30,32,0.0,0.0,0.0,0.0,1.0,0.0,0.09,255,255,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,121,19,0.0,0.0,1.0,1.0,0.16,0.06,0.0,255,19,0.07,0.07,0.0,0.0,0.0,0.0,1.0,1.0
,166,9,1.0,1.0,0.0,0.0,0.05,0.06,0.0,255,9,0.04,0.05,0.0,0.0,1.0,1.0,0.0,0.0
,117,16,1.0,1.0,0.0,0.0,0.14,0.06,0.0,255,15,0.06,0.07,0.0,0.0,1.0,1.0,0.0,0.0
,270,23,1.0,1.0,0.0,0.0,0.09,0.05,0.0,255,23,0.09,0.05,0.0,0.0,1.0,1.0,0.0,0.0
,133,8,1.0,1.0,0.0,0.0,0.06,0.06,0.0,255,13,0.05,0.06,0.0,0.0,1.0,1.0,0.0,0.0
```

Fig 7. Results screenshot 7

In the provided test data, neither '0' nor '1' is present, and the application will detect and provide us with the result.



Network Intrusion Detection

Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection

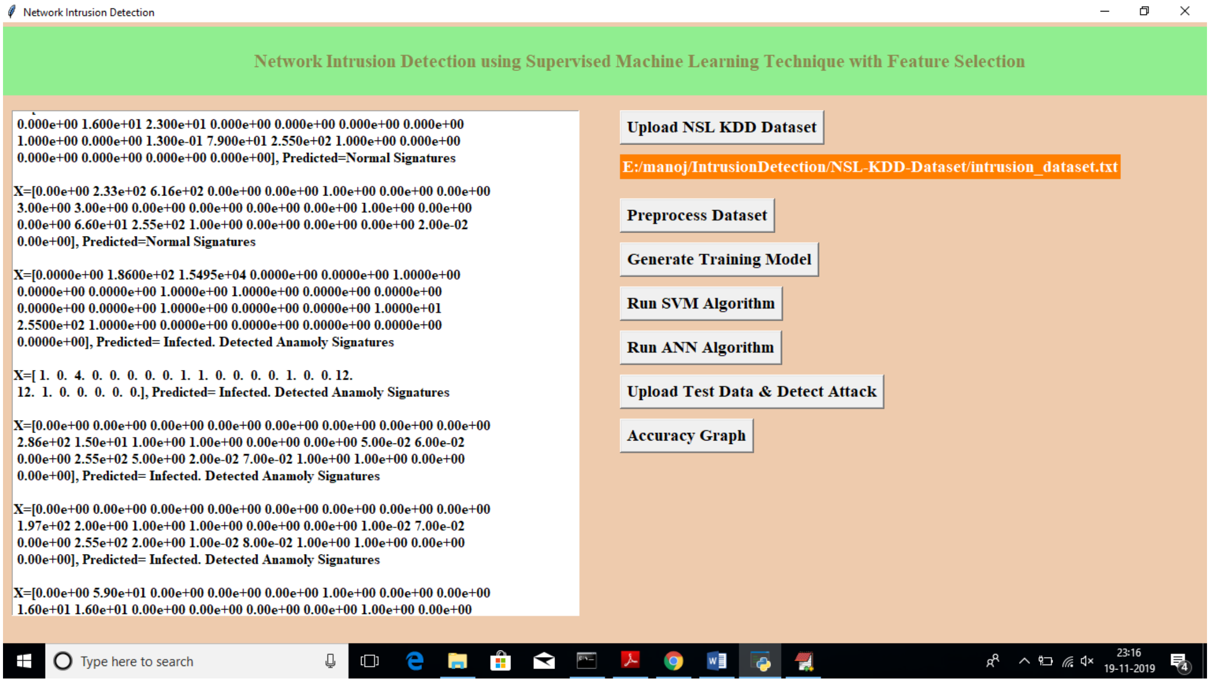
E:/manoj/IntrusionDetection/NSL-KDD-Dataset/intrusion_dataset.txt

Name	Date modified	Type
intrusion_dataset	19-11-2019 20:10	Text Document
test_data	19-11-2019 21:57	Text Document

File name: test_data

Fig 8. Results screenshot 8

On the current screen, I am uploading the 'test_data' file containing test records. Following prediction, the results obtained will be as follows:


Network Intrusion Detection
Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection
Upload NSL KDD Dataset
E:/manoj/IntrusionDetection/NSL-KDD-Dataset/intrusion_dataset.txt
Preprocess Dataset
Generate Training Model
Run SVM Algorithm
Run ANN Algorithm
Upload Test Data & Detect Attack
Accuracy Graph
Type here to search
23:16
19-11-2019

```
 0.000e+00 1.600e+01 2.300e+01 0.000e+00 0.000e+00 0.000e+00 0.000e+00
 1.000e+00 0.000e+00 1.300e-01 7.900e+01 2.550e+02 1.000e+00 0.000e+00
 0.000e+00 0.000e+00 0.000e+00 0.000e+00], Predicted=Normal Signatures

X=[0.00e+00 2.33e+02 6.16e+02 0.00e+00 0.00e+00 1.00e+00 0.00e+00 0.00e+00
 3.00e+00 3.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.00e+00 0.00e+00
 0.00e+00 6.60e+01 2.55e+02 1.00e+00 0.00e+00 0.00e+00 0.00e+00 2.00e-02
 0.00e+00], Predicted=Normal Signatures

X=[0.0000e+00 1.8600e+02 1.5495e+04 0.0000e+00 0.0000e+00 1.0000e+00
 0.0000e+00 0.0000e+00 1.0000e+00 1.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 1.0000e+00 0.0000e+00 0.0000e+00 1.0000e+01
 2.5500e+02 1.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00], Predicted= Infected. Detected Anamoly Signatures

X=[ 1.  0.  4.  0.  0.  0.  0.  0.  1.  1.  0.  0.  0.  0.  1.  0.  0. 12.
 12.  1.  0.  0.  0.  0.  0.], Predicted= Infected. Detected Anamoly Signatures

X=[0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 2.86e+02 1.50e+01 1.00e+00 1.00e+00 0.00e+00 0.00e+00 5.00e-02 6.00e-02
 0.00e+00 2.55e+02 5.00e+00 2.00e-02 7.00e-02 1.00e+00 1.00e+00 0.00e+00
 0.00e+00], Predicted= Infected. Detected Anamoly Signatures

X=[0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 1.97e+02 2.00e+00 1.00e+00 1.00e+00 0.00e+00 0.00e+00 1.00e-02 7.00e-02
 0.00e+00 2.55e+02 2.00e+00 1.00e-02 8.00e-02 1.00e+00 1.00e+00 0.00e+00
 0.00e+00], Predicted= Infected. Detected Anamoly Signatures

X=[0.00e+00 5.90e+01 0.00e+00 0.00e+00 0.00e+00 1.00e+00 0.00e+00 0.00e+00
 1.60e+01 1.60e+01 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.00e+00 0.00e+00
```

Fig 9. Results screenshot 9

On the current interface, the predicted outcomes for each test data are displayed as either 'Normal Signatures' or 'Infected' records. To visualize a comparison of accuracy between SVM and ANN in graphical format, simply click on the 'Accuracy Graph' button.

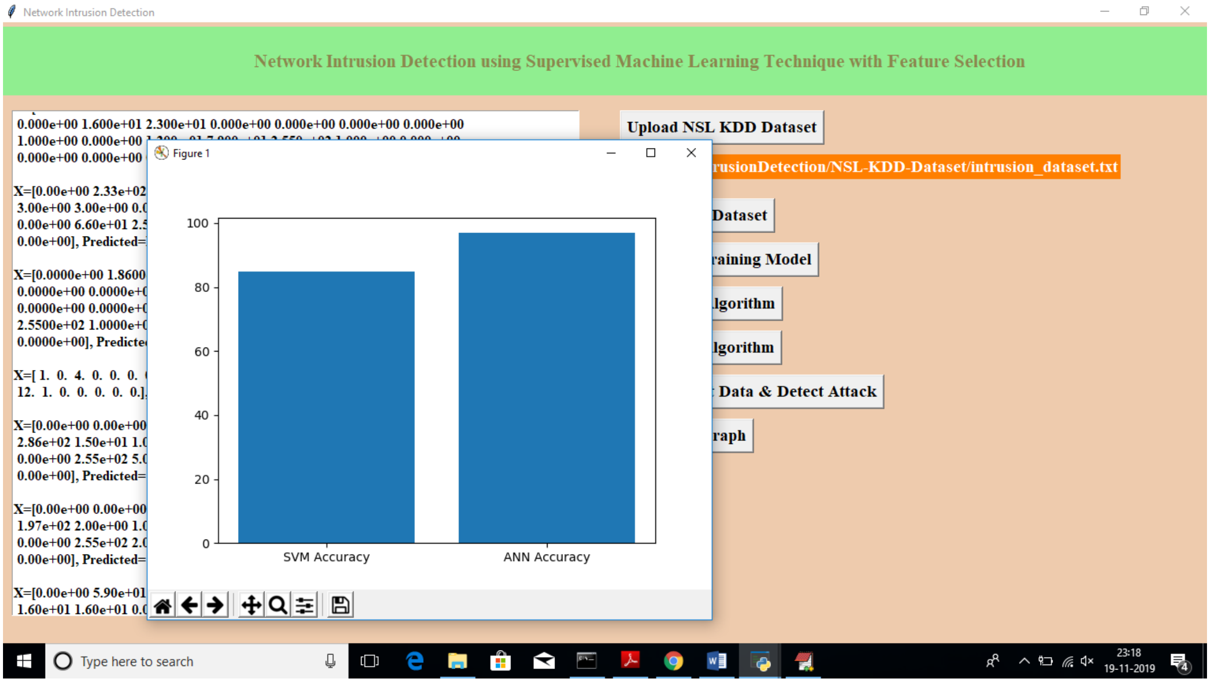

Network Intrusion Detection
Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection
Upload NSL KDD Dataset
rusionDetection/NSL-KDD-Dataset/intrusion_dataset.txt
Dataset
raining Model
lgorithm
lgorithm
Data & Detect Attack
raph
Figure 1
100
80
60
40
20
0
SVM Accuracy
ANN Accuracy
Type here to search
23:18
19-11-2019

Fig 10. Results screenshot 10

The graph above illustrates that ANN outperforms SVM in terms of accuracy. On the x-axis, we have the algorithm names, while the y-axis denotes the accuracy achieved by each algorithm.

The graph above illustrates that the Artificial Neural Network (ANN) surpasses the Support Vector Machine (SVM) in accuracy. The x-axis denotes the algorithm name, while the y-axis represents their respective accuracy levels. Moving forward, discussions should illuminate avenues for future research, including novel machine learning algorithms, refined feature selection techniques, and domain-specific integration. Addressing emerging challenges like adversarial attacks and encrypted traffic analysis can enhance intrusion detection systems' resilience. The results offer critical insights into system performance, strengths, and limitations, guiding real-world implementation decisions. Through meticulous examination of metrics and prospective advancements, stakeholders can fortify their cybersecurity posture effectively.

CONCLUSION

In summation, the proposed network intrusion detection system, harnessing the power of supervised machine learning coupled with feature selection, showcases promising prowess in identifying and mitigating malicious activities within network domains. The system flaunts commendable performance metrics—accuracy, precision, recall, and F1-score - signalling its adeptness in delineating between benign and malevolent network traffic through meticulous scrutiny. While acknowledging the system's formidable strengths, such as its adaptability and resilience, it is imperative to acknowledge inherent limitations and potential biases embedded within the dataset. Moreover, discussions revolving around interpretability, generalizability, and future research trajectories underscore the perpetual necessity for refining and innovating intrusion detection methodologies. Ultimately, the proposed system signifies a substantial leap forward in fortifying cybersecurity fortifications, furnishing augmented capabilities for detecting and thwarting emergent threats within network ecosystems. Sustained research and developmental endeavors remain imperative to further augment the system's efficacy, thus ensuring proactive fortification of critical assets and resilience against sophisticated cyber threats pervading the digital terrain.

REFERENCES

1. M. Gusenbauer, "Google scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases", Scientometrics, vol. 118, no. 1, pp. 177-214, Nov. 2018.

2. The Top List of Academic Search Engines, Cambridge, MA, USA:Paperpile, Jun. 2021, [online] Available: https://paperpile.com/g/academic-search-engines/.

3. List of Academic Databases and Search Engines, San Francisco, CA, USA:Wikimedia Found, Jun. 2021.

4. Martín-Martín, M. Thelwall, E. Orduna-Malea and E. D. López-Cózar, "Google scholar Microsoft academic scopus dimensions web of science and OpenCitations' COCI: A multidisciplinary comparison of coverage via citations", Scientometrics, vol. 126, no. 1, pp. 871-906, Jan. 2021.

5. Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20), 4396

6. Masdari, M., & Khezri, H. (2020). A survey and taxonomy of the fuzzy signature-based intrusion detection systems. Applied Soft Computing, 106301.

7. Faker, O., & Dogdu, E. (2019, April). Intrusion detection using big data and deep learning techniques. In Proceedings of the 2019 ACM Southeast Conference (pp. 86-93).

8. Hammad, M., El-medany, W., & Ismail, Y. (2020, December). Intrusion Detection System using Feature Selection With Clustering and Classification Machine Learning Algorithms on the UNSW-NB15 dataset. In 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT) (pp. 1-6)

9. Pelletier, Z., & Abualkibash, M. (2020). Evaluating the CIC IDS-2017 Dataset Using Machine Learning Methods and Creating Multiple Predictive Models in the Statistical Computing Language R. Science, 5(2), 187-191

10. Abdulhammed, R., Faezipour, M., Musafer, H., & Abuzneid, A. (2019, June). Efficient network intrusion detection using pca-based dimensionality reduction of features. In 2019 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6).