

FLEXIBLE MACHINE LEARNING BASED CYBERATTACK DETECTION USING SPATIOTEMPORAL PATTERNS FOR DISTRIBUTION SYSTEMS

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

KANDIBANDA SRIKANTH (Reg. No. 38110230)

GUDIPATI NAGARAJU (Reg. No. 38110176)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL- 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KANDIBANDA SRIKANTH (Reg. No: 38110230), GUDIPATI NAGARAJU (Reg. No: 38110176)** who carried out the project entitled "**FLEXIBLE MACHINE LEARNING BASED CYBERATTACK DETECTION USING SPATIOTEMPORAL PATTERNS FOR DISTRIBUTION SYSTEMS**" under my supervision from November 2021 to April 2022.

Internal Guide

Dr.S. PRINCE MARY M.E., Ph.D.,

Head of the Department

Dr. L. LAKSHMANAN M.E., Ph.D.,

Dr. S. VIGNESHWARI M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **KANDIBANDA SRIKANTH, GUDIPATI NAGARAJU** hereby declare that the Project Report entitled **FLEXIBLE MACHINE LEARNING BASED CYBERATTACK DETECTION USING SPATIO-TEMPORAL PATTERNS FOR DISTRIBUTION SYSTEMS** done by me under the guidance of **Dr. S. PRINCE MARY M.E., Ph.D., (Internal)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 09-04-2022

K. Srikanth

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing** **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr. S. Vigneshwari M.E., Ph.D. Heads of the Department, Dept. of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Prince Mary ME., Ph.D** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In smart cities electricity consumption will be monitor by sensors and send this details to energy distributed systems and this distributed system will adjust energy supply based on sensor data. Sometimes attackers may hack the sensors and send wrong information to the distributed system it leads to the problem for distributed system. To overcome this problem GGL and Flexible Bayes Classifiers are used.

Data is passed to the GGL algorithm Which gives Laplacian matrix . Laplacian matrix can calculate it is normal or abnormal values. Data obtained from GGL will be pass as input to the Bayes Classifier and this will be trained on GGL values and it generate a classifier model.

This project develops a flexible machine learning detection method for cyberattacks in distribution systems considering spatiotemporal patterns. Spatiotemporal patterns are recognized by the graph Laplacian based on system-wide measurements. A flexible Bayes classifier (BC) is used to train spatiotemporal patterns which could be violated when cyberattacks occur. Cyberattacks are detected by using flexible BCs online. The effectiveness of the developed method is demonstrated through standard IEEE 13- and 123-node test feeders.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTRACT	i
	LIST OF FIGURES	iv
	LIST OF ABBREVIATIONS	v
1	INTRODUCTION	1
	1.1 Motivation	2
	1.2 Problem definition	2
	1.3 Objective of project	2
2	LITERATURE SURVEY	4
	2.1 LITERATURE SURVEY	4
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	8
	3.1 Scope	8
	3.2 Limitations of project	8
	3.3 Existing system	8
	3.3.1 Disadvantages	9
	3.4 Proposed work	9
	3.4.1 Advantage	9
4	METHODS AND ALGORITHMS USED	10
	4.1 Software and Hardware Requirements	10
	4.1.1 Software Requirements	10
	4.1.2 Hardware Requirements	10
	4.2 System architecture	10
	4.3 Data flow diagram	11
	4.4 Module Implementation	12

	4.4.1 Upload spatiotemporal load dataset	12
	4.4.2 Data preprocess	13
	4.4.3 Run General Graph Laplacian	15
	4.4.4 Model Generation	15
	4.4.5 Run SVM Algorithm	15
	4.4.6 Run Naive Bayes Algorithm	16
	4.4.7 Run Propose Flexible Bayes Classifier	18
	4.4.8 True Prediction Rate Graph	18
5	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	20
6	SUMMARY AND CONCLUSIONS	28
	REFERENCES	29
	APPENDIX	30
	A. SOURCE CODE	30
	B. SCREENSHOTS	35
	C. RESEARCH PAPER	38

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	System Architecture	11
4.2	Data Flow Diagram	12
4.3	SVM Algorithm	16
4.4	Naive Bayes Algorithm	18
4.5	Comparing Algorithms	19
5.1	GGL Matrix Calculation	22
5.2	Home Page	22
5.3	File Upload	23
5.4	File Uploaded	23
5.5	GGL Graph	24
5.6	Splitting the Dataset	25
5.7	SVM Accuracy	25
5.8	Naive Bayes, Flexible Bayes Accuracy	26
5.9	Comparing Algorithms	27
6.1	Home Page of Application	35
6.2	Loading Dataset	35
6.3	Network Graph	36
6.4	Accuracy Rates of Algorithms	36
6.5	Comparing Algorithms	37

LIST OF ABBREVIATIONS

OOP	Object-Oriented Programming
PIP	Package Installer for Python
NLTK	Natural Language Toolkit
SVM	Support Vector Machine

CHAPTER - 1

INTRODUCTION

The growing deployment of distributed energy resources (DERs), microgrids, and other distribution-level technology and assets has completely changed the way the distribution systems have been designed and operated traditionally. Also, as an increasing number of sensors (e.g., micro-PMUs) are being developed and deployed on the distribution system in conjunction with the conventional SCADA systems, advanced metering infrastructure (AMI), and other field devices to enable data-driven observability and gridedge data analytics [1], the attack surface to the distribution management system (DMS) is inevitably enlarged. DMS and associated monitoring and control systems are among the key actors for making decisions and exchanging information.

However, existing cybersecurity technologies employed in distribution systems are still vulnerable to cyberattacks. It is highly necessary to develop cyber-resilient DMS functions and cybersecurity technologies to enable future energy delivery systems to accurately detect, dynamically adapt, successfully survive and reject a cyberattack. Unlike conventional cyberattack detection techniques, such as naive Bayes classifiers (BCs) which are highly based on the normality assumption, this letter attempts to capture the continuous attribute of spatiotemporal patterns among system measurements by developing flexible BCs. Essentially, spatiotemporal patterns of measurement data under normal conditions would be compromised when cyberattacks occur.

Based on this concept, this letter seeks to address two critical questions for the cyberattack detection on distribution systems. (i) Is it possible to quantitatively capture the spatiotemporal patterns between cyberattack scenarios and normal scenarios? (ii) Can operators deploy flexible BCs to enhance the accuracy of conventional cyberattack detection methods? In this letter, we seek to integrate the spatiotemporal patterns of system measurements into a flexible BC for cyberattack detection. Specifically, spatiotemporal patterns are captured by the generalized graph Laplacian (GGL) matrix for system measurements. For the training process of the proposed

flexible BC, they are taken as its input variables, while the labels of cyberattack templates are taken as its output variables. For the testing process, the online spatiotemporal patterns captured by GGL are put into the proposed flexible BC, which subsequently outputs the cyberattack detection results.

1.1 MOTIVATION:

This letter develops a novel anomaly detection method using the generalized graph Laplacian (GGL) matrix to visualize the spatiotemporal relationship of distribution-level phasor measurement unit (μ PMU) data. The μ PMU data in a specific time horizon are segregated into multiple segments. An optimization problem formulated as a Lagrangian function is utilized to estimate the GGL matrix. During the iterative process, an optimal update is constituted as a quadratic program problem. To perform the μ PMU-based spatiotemporal analysis, normalized diagonal elements of GGL matrix are proposed as a quantitative metric. The effectiveness of the developed method is demonstrated through real-world μ PMU measurements gathered from test feeders in Riverside, CA, USA.

1.2 PROBLEM DEFINITION:

Similarly, image and video signals can be modeled using weighted graphs whose weights capture the correlation or similarity between neighboring pixel values (such as in nearest-neighbor models) [5]–[10]. In practice, many datasets typically consist of an unstructured list of samples, where the associated graph information (representing the structural relations between samples/features) is latent. For analysis, processing and algorithmic purposes, it is often useful to build graph-based models that provide a concise/simple explanation for datasets and also reduce the dimension of the problem by exploiting the available prior knowledge/assumptions about the desired graph (e.g., structural information including connectivity and sparsity level).

1.3 OBJECTIVE OF PROJECT:

The focus of this paper is on learning graphs (i.e., graph-based models) from data, where the basic goal is to optimize a weighted graph with nonnegative edge

weights characterizing the affinity relationship between the entries of a signal/data vector based on multiple observed vectors. For this purpose, we propose a general framework where graph learning is formulated as the estimation of different types of graph Laplacian matrices from data.

CHAPTER - 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

[1] An Optimized Swinging Door Algorithm for Identifying Wind Ramping Events:

Authors: Mingjian Cui, Student Member, IEEE, Jie Zhang, Senior Member, IEEE, Anthony R. Florita, Member, IEEE, Bri-Mathias Hodge, Member, IEEE, Deping Ke, and Yuanzhang Sun, Senior Member, IEEE

Abstract:

With the increasing penetration of renewable energy in recent years, wind power ramp events (WPRES) have started affecting the economic and reliable operation of power grids. In this paper, we develop an optimized swinging door algorithm (OpSDA) to improve the state of the art in WPRES detection. The swinging door algorithm (SDA) is utilized to segregate wind power data through a piecewise linear approximation. A dynamic programming algorithm is performed to optimize the segments by: 1) merging adjacent segments with the same ramp changing direction; 2) handling wind power bumps; and 3) postprocessing insignificant-ramps intervals. Measured wind power data from two case studies are utilized to evaluate the performance of the proposed OpSDA. Results show that the OpSDA provides 1) significantly better performance than the SDA and 2) equal-to-better performance compared to the L1-Ramp Detect with Sliding Window (L1-SW) method with significantly less computational time.

[2] Machine Learning-Based Anomaly Detection for Load Forecasting Under Cyberattacks:

Authors: Mingjian Cui , Senior Member, IEEE, Jianhui Wang , Senior Member, IEEE, and Meng Yue , Member, IEEE

Abstract:

Accurate load forecasting can create both economic and reliability benefits for power system operators. However, the cyberattack on load forecasting may mislead operators to make unsuitable operational decisions for the electricity delivery. To effectively and accurately detect these cyberattacks, this paper develops a machine learning-based anomaly detection (MLAD) methodology. First, load forecasts provided by neural networks are used to reconstruct the benchmark and scaling data by using the k-means clustering. Second, the cyberattack template is estimated by the naive Bayes classification based on the cumulative distribution function and statistical features of the scaling data. Finally, the dynamic programming is utilized to calculate both the occurrence and parameter of one cyberattack on load forecasting data. A widely used symbolic aggregation approximation method is compared with the developed MLAD method. Numerical simulations on the publicly load data show that the MLAD method can effectively detect cyberattacks for load forecasting data with relatively high accuracy. Also, the robustness of MLAD is verified by thousands of attack scenarios based on Monte Carlo simulation.

[3] Graph Learning From Data Under Laplacian and Structural Constraints:

Authors: Hilmi E. Egilmez, Student Member, IEEE, Eduardo Pavez, Student Member, IEEE, and Antonio Ortega, Fellow, IEEE

Abstract:

Graphs are fundamental mathematical structures used in various fields to represent data, signals, and processes. In this paper, we propose a novel framework for learning/estimating graphs from data. The proposed framework includes (i) formulation of various graph learning problems, (ii) their probabilistic interpretations, and (iii) associated algorithms. Specifically, graph learning problems are posed as the estimation of graph Laplacian matrices from some observed data under given structural constraints (e.g., graph connectivity and sparsity level). From a probabilistic perspective, the problems of interest

correspond to maximum a posteriori parameter estimation of Gaussian–Markov random field models, whose precision (inverse covariance) is a graph Laplacian matrix. For the proposed graph learning problems, specialized algorithms are developed by incorporating the graph Laplacian and structural constraints. The experimental results demonstrate that the proposed algorithms outperform the current state-of-the-art methods in terms of accuracy and computational efficiency.

[4] Generalized Graph Laplacian Based Anomaly Detection for Spatiotemporal MicroPMU Data:

Authors: Mingjian Cui , Senior Member, IEEE, Jianhui Wang , Senior Member, IEEE, Anthony R. Florita , Member, IEEE, and Yingchen Zhang , Senior Member, IEEE

Abstract:

This letter develops a novel anomaly detection method using the generalized graph Laplacian (GGL) matrix to visualize the spatiotemporal relationship of distribution-level phasor measurement unit (μ PMU) data. The μ PMU data in a specific time horizon are segregated into multiple segments. An optimization problem formulated as a Lagrangian function is utilized to estimate the GGL matrix. During the iterative process, an optimal update is constituted as a quadratic program problem. To perform the μ PMU-based spatiotemporal analysis, normalized diagonal elements of GGL matrix are proposed as a quantitative metric. The effectiveness of the developed method is demonstrated through real-world μ PMU measurements gathered from test feeders in Riverside, CA, USA.

[5] Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges:

Authors: Yi Wang, Qixin Chen, Tao Hong, Chongqing Kang

Abstract:

The widespread popularity of smart meters enables an immense amount of fine-

grained electricity consumption data to be collected. Meanwhile, the deregulation of the power industry, particularly on the delivery side, has continuously been moving forward worldwide. How to employ massive smart meter data to promote and enhance the efficiency and sustainability of the power grid is a pressing issue. To date, substantial works have been conducted on smart meter data analytics. To provide a comprehensive overview of the current research and to identify challenges for future research, this paper conducts an application-oriented review of smart meter data analytics. Following the three stages of analytics, namely, descriptive, predictive and prescriptive analytics, we identify the key application areas as load analysis, load forecasting, and load management. We also review the techniques and methodologies adopted or developed to address each application. In addition, we also discuss some research trends, such as big data issues, novel machine learning technologies, new business models, the transition of energy systems, and data privacy and security.

CHAPTER - 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 SCOPE:

To develop a novel μ PMU-based anomaly detection method by using a generalized graph Laplacian (GGL) matrix. The developed detection method utilizes a Laplacian optimization function to estimate the GGL matrix. A quadratic program technique is used to design the optimal update during the computation process.

3.2 LIMITATIONS OF PROJECT

To provide an overview of the existing research in smart meter data analytics, a bibliometric analysis was conducted on 31 December 2017 using the well-established and acknowledged databases, Web of Science (WoS). The query for WoS is as follows: TS=((“smart meter” OR “consumption” OR “demand” OR “load”) AND “data” AND (“household” OR “resident” OR “residential” OR “building” OR “industrial” OR “individual” OR “customer” OR “consumer”) AND (“energy theft” OR “demand response” OR “clustering” OR “forecasting” OR “profiling” OR “classification” OR “abnormal” OR “anomaly”) AND (“smart grid” OR “power system”)).

3.3 EXISTING SYSTEM:

Existing cybersecurity technologies employed in distribution systems are still vulnerable to cyberattacks. It is highly necessary to develop cyber-resilient DMS functions and cybersecurity technologies to enable future energy delivery systems to accurately detect, dynamically adapt, successfully survive and reject a cyberattack. Unlike conventional cyberattack detection techniques, such as naive Bayes classifiers (BCs) which are highly based on the normality assumption, this letter attempts to capture the continuous attribute of spatiotemporal patterns among system

measurements by developing flexible BCs. Essentially, spatiotemporal pattern of measurement data under normal conditions would be compromised when cyberattacks occur.

3.3.1 Disadvantage:

- Accuracy is less

3.4 PROPOSED WORK:

This project develops a flexible machine learning detection method for cyberattacks in distribution systems considering spatiotemporal patterns. Spatiotemporal patterns are recognized by the graph Laplacian based on system-wide measurements. A flexible Bayes classifier (BC) is used to train spatiotemporal patterns which could be violated when cyberattacks occur.

3.4.1 Advantage:

- Accuracy is high

CHAPTER - 4

METHODS AND ALGORITHMS USED

4.1 SOFTWARE AND HARDWARE REQUIREMENTS

4.1.1 Software Requirements

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Python idle 3.7 version (or)**
- **Anaconda 3.7 (or)**
- **Jupyter (or)**
- **Google Colaboratory**

4.1.2 Hardware Requirements

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Operating system** : **windows, Linux**
- **Processor** : **minimum intel i3**
- **Ram** : **minimum 4gb**

- Hard disk : minimum 250gb

4.2 SYSTEM ARCHITECTURE:

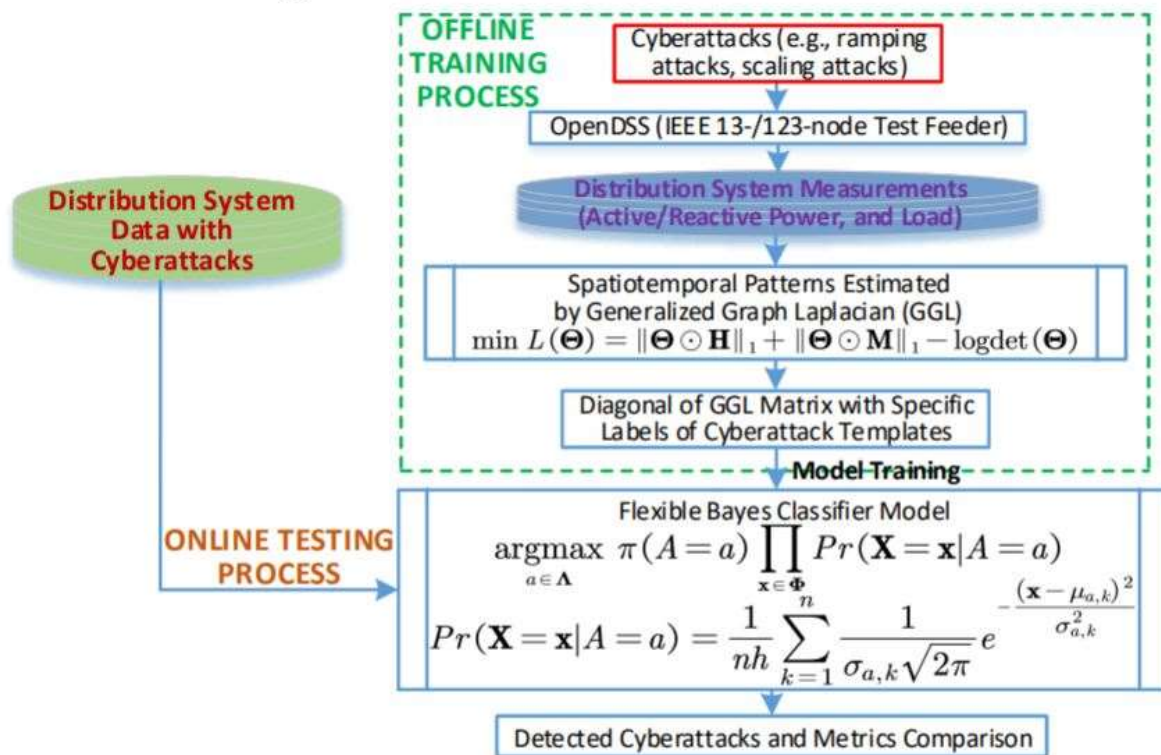


Fig:4.1 System Architecture

4.3 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts

information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

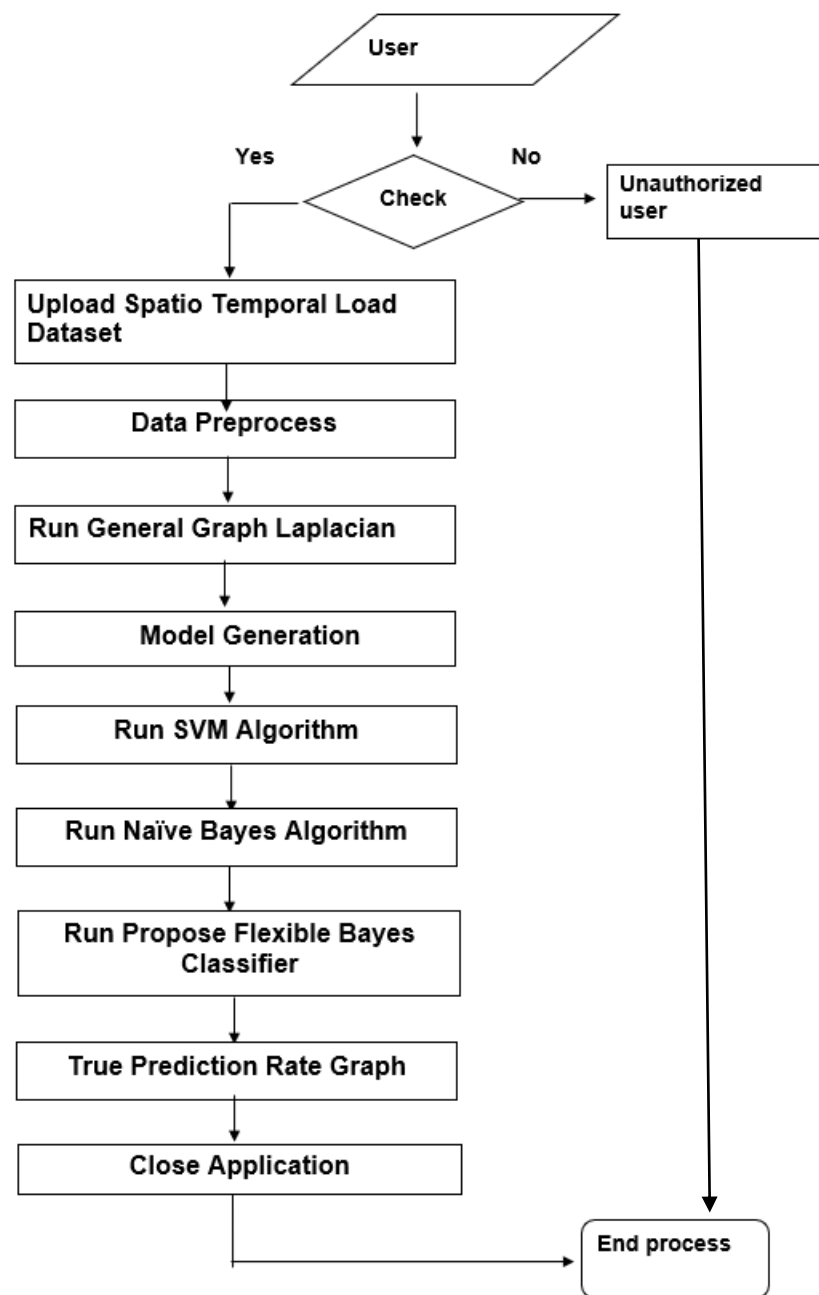


Fig: 4.2 Data flow diagram

4.4 MODULE IMPLEMENTATION:

4.4.1 Upload SpatioTemporal Load Dataset:

Spatiotemporal data which consists of sensor's location, date, time, etc., The ability to effectively estimate expected demand is one of the most important criteria in power grid operation and planning. Grid operations, energy management, and planning can all be improved as a result of this. Load forecasting has traditionally relied on aggregated spatial and temporal consumption data; however, with the implementation of Advanced Metering Infrastructure systems, it is now possible to do so not just at the system level, but also at the customer level. Novel ideas and methods for load forecasting at a finer level can be investigated with this new influx of data. In this paper, a novel k-nearest Vector Autoregressive framework with exogenous input is suggested to estimate household-level electricity demand spatially and temporally from extremely short to mid-term. Thousands of residential and commercial households' smart metre time series and geographic data were analysed.

4.4.2 Data Preprocess:

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can

add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose & read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame

- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

4.4.3 Run General Graph Laplacian:

The Laplacian establishes a natural relationship between discrete and continuous representations, such as graphs and vector spaces and manifolds. The Laplacian's most well-known use is spectral clustering, which is a computationally tractable solution to the graph partitioning problem. While Laplacians can be described for a variety of spaces, data is typically in the form of a set of connected that may or may not be prepared to lie in the area of interest, and must frequently be guessed. Many techniques begin by estimating the Laplacian or some of its features, such as its eigenvalues and eigenfunctions, as well as maybe certain functions formed from these functions, and these items are taken to determine the space of interest.

4.4.4 Model Generation:

Model generation is done using electrical data consumption details sent by sensors, and we can observe time details (here, time is a node) coupled to energy consumption as edges. It will derive Laplacian values from the model, which will be fed into the Bayes classifier, and his data is generated by GGL.

4.4.5 Run SVM Algorithm:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can

easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

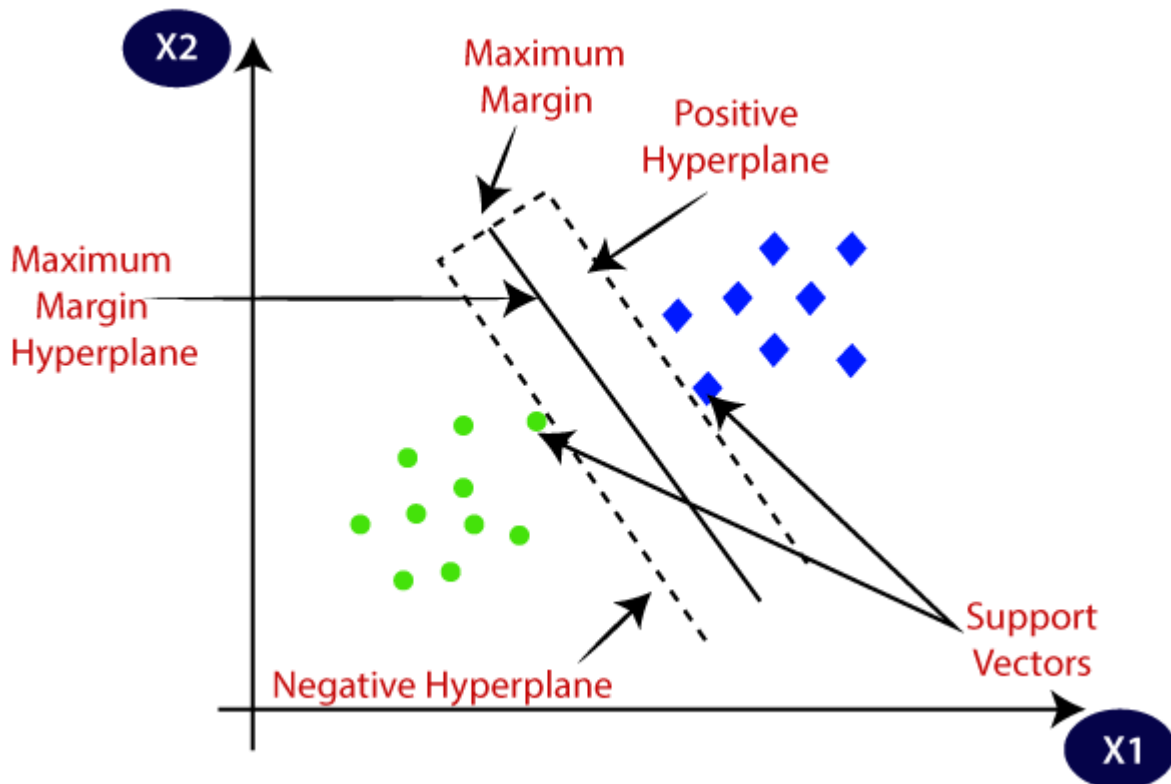


Fig: 4.3 SVM Algorithm

4.4.6 Run Naïve Bayes Algorithm:

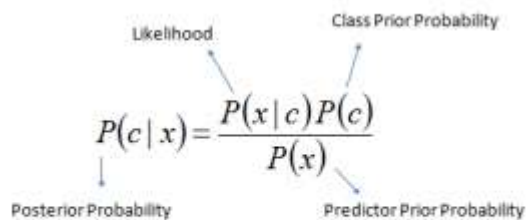
It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon

the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$. Look at the equation below:



$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Above,

- $P(c|x)$ is the posterior probability of *class* (c , *target*) given predictor (x , attributes)
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

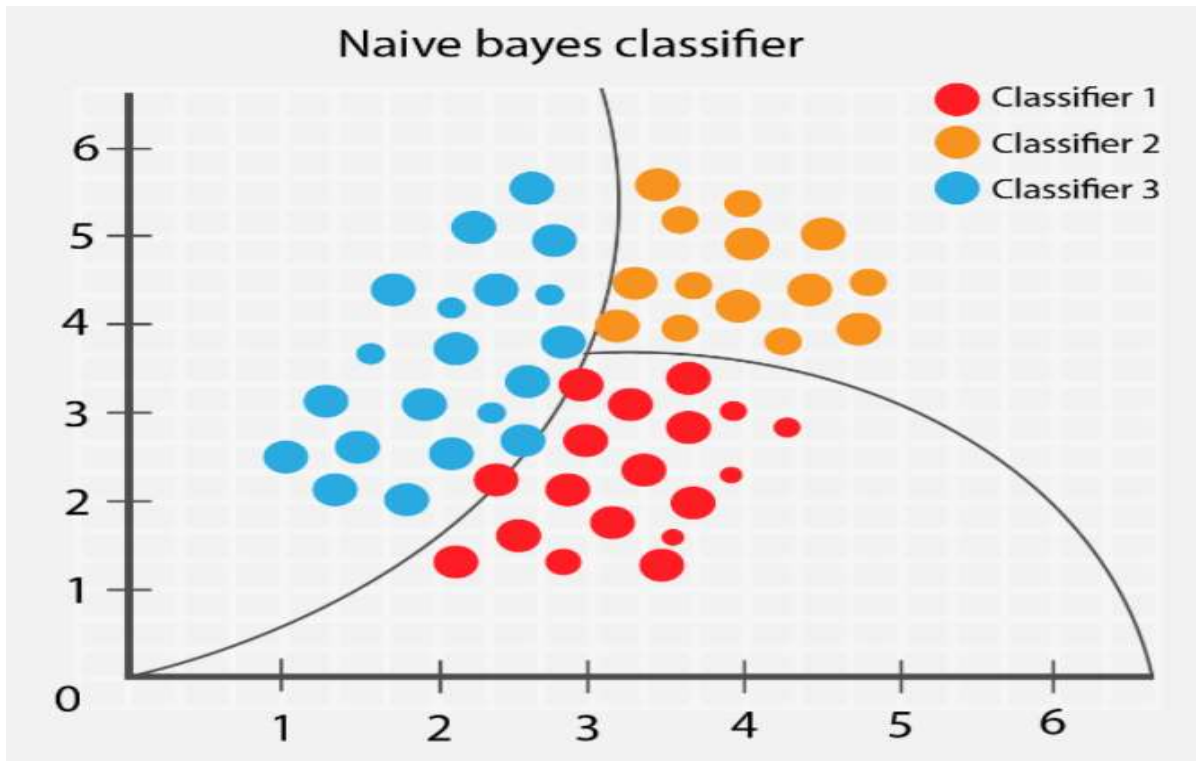


Fig: 4.4 Naïve Bayes Algorithm

4.4.7 Run Propose Flexible Bayes Classifier:

In statistical classification, the Bayes classifier is a valuable benchmark. The excess risk of a general classifier is defined as perhaps based on some training data. As a result, this non-negative variable is critical for evaluating the effectiveness of various classification techniques.

4.4.8 True Prediction Rate Graph:

The x-axis in the above graph shows the algorithm name, while the y-axis represents the TPR values of those algorithms. From the graph that Flexible Bayes classifiers have a higher prediction rate than existing SVM and Naive Bayes algorithms.

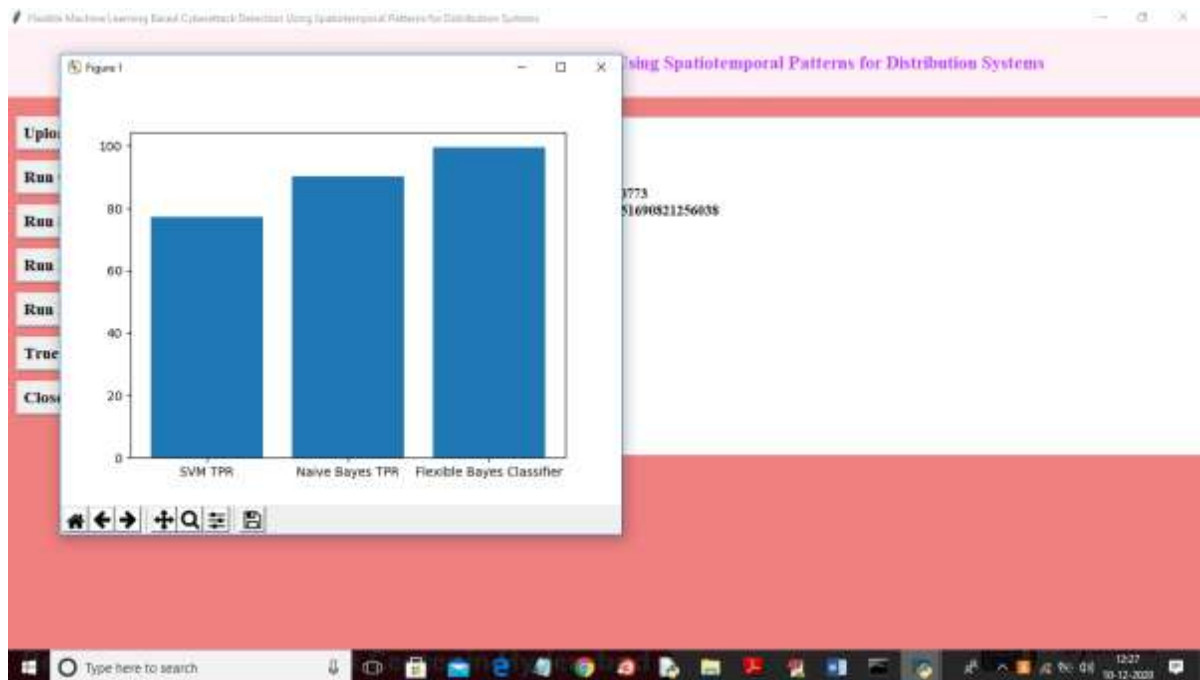


Fig: 4.5 Comparing Algorithms

- Close Application

CHAPTER – 5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

Now-a-days in smart cities electricity consumption will be monitor by sensors and send this details to energy distributed systems and this distributed system will adjust energy supply based on sensor data. For example if sensor sense more energy consumption for particular home then distributed system will reserve more energy for that home. So distributed systems will take decision based on sensors input and sometime some attackers may hack sensors and then modify sensors value and this modified data send to distributed systems and this system may take wrong decisions.

Modification of sensors data by attackers create different types of attack such as SCALING, RAMPING, RANDOM and SMOOTH CURVE attack. Here Spatiotemporal Patterns refers to sensors data which contain information of sensor place, date and time. Any data which contains PLACE and DATE information is called as Spatiotemporal Patterns.

Existing machine learning algorithms may not detect above attacks which is coming from online sensors and to overcome from this problem author is using GGL (General Graph Laplacian) and Flexible Bayes classifier.

Propose concept work in two steps

- 1) In this step GGL algorithm will be applied on sensors historical data and this algorithm will create a matrix and then calculate similarity matrix called W and then calculate Degree Matrix called D . Subtraction of D and W will give us laplacian matrix.

W = Similarity Matrix

D = Degree Matrix

$L = D - W$

If the laplacian matrix row contains negative values then sensor data contains abnormal values and this data will be marked as CYBER attack and if matrix row does not contains any negative values then that

row will be consider as normal.

- 2) Flexible Bayes Classifier: Data obtained from GGL will be pass as input to the Bayes classifier and then this classifier will be trained on GGL input to generate a classifier model. Classifier model will be applied on new test data to predict whether sensor data contains normal to attack signatures. Naïve Bayes algorithm work similar to human brain as human brain take decision based on satisfying condition and Naïve Bayes will do same work.

For example: to allow a child to play outside then human will check below condition

If temperature is not hot AND not raining {

Then allow child to play

} else {

Not allowed

}

To implement this project we have use load dataset from GRID energy distributed system and this dataset saved inside dataset folder. In this paper author have done performance comparison between SVM, existing Naïve Bayes and propose Flexible Bayes Classifier. In propose Bayes classifier we are getting input from GGL so author gave it name as Bayes Classifier.

Below code screen shots displaying how we have calculated GGL values

In above screen click on 'Upload SpatioTemporal Load Dataset' button to load dataset

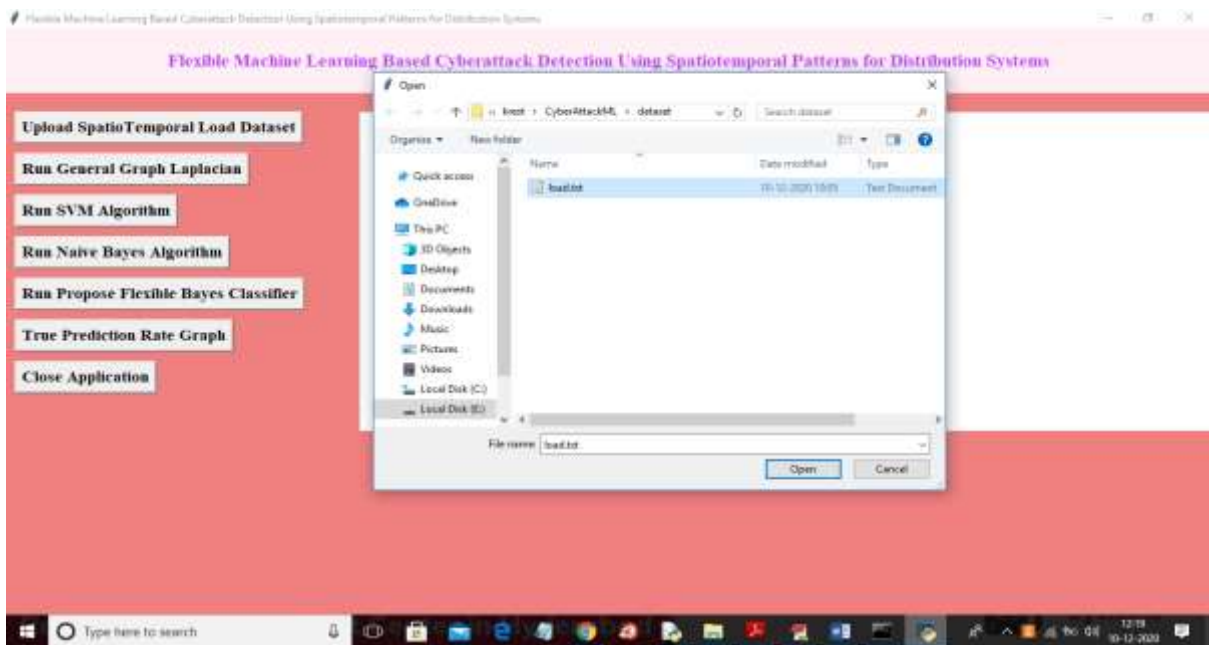


Fig: 5.3 File Upload

In above screen selecting and uploading 'load.txt' file and then click on 'Open' button to load dataset and to get below screen



Fig: 5.4 File uploaded

In above screen dataset is loaded and now click on 'Run General Graph Laplacian' button to calculate laplacian matrix and to get below graph

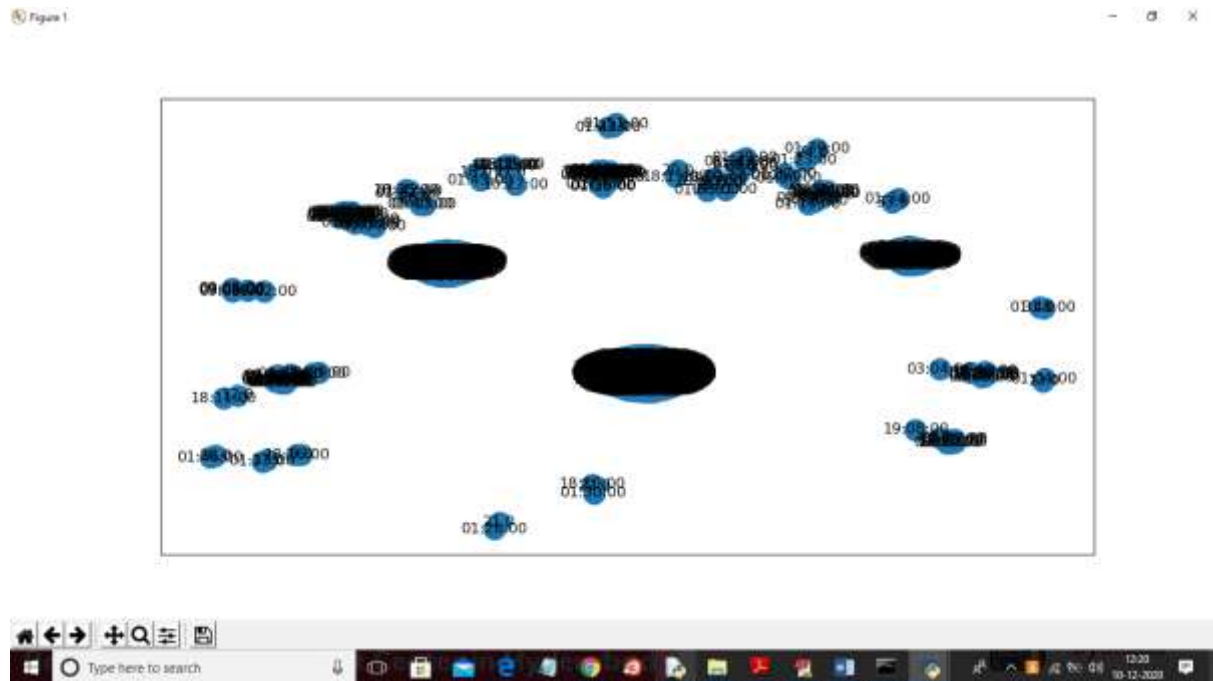


Fig: 5.5 GGL Graph

Above graph generated based on electricity data consumption details send by sensors and in above graph we can see time details (here time is node) connected as edges to energy consumption. From above graph we will calculate laplacian values which will be input to Bayes classifier and below screen we can see input data generated from GGL

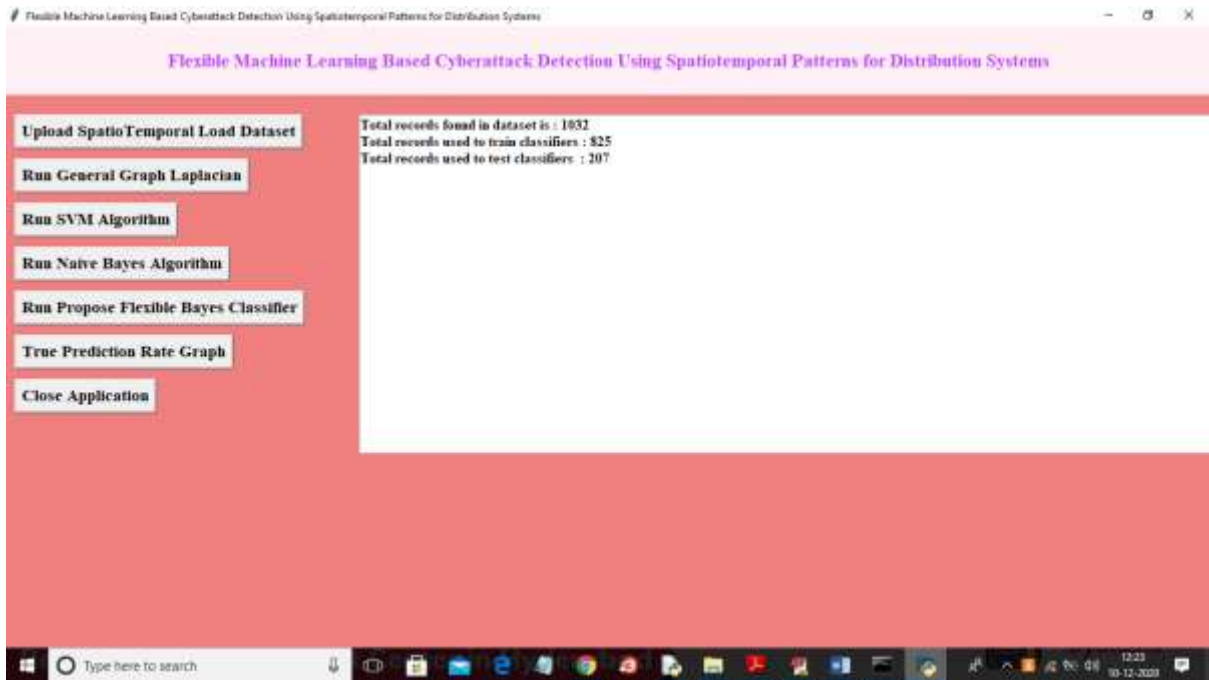


Fig: 5.6 Splitting the Dataset

In above screen we can see application got total 1032 GGL records and application using 825 records to train Bayes classifier and 267 records to test or calculate Bayes classifier true prediction rate. Now input data is ready and now click on 'Run SVM Algorithm' button to get its TPR value.

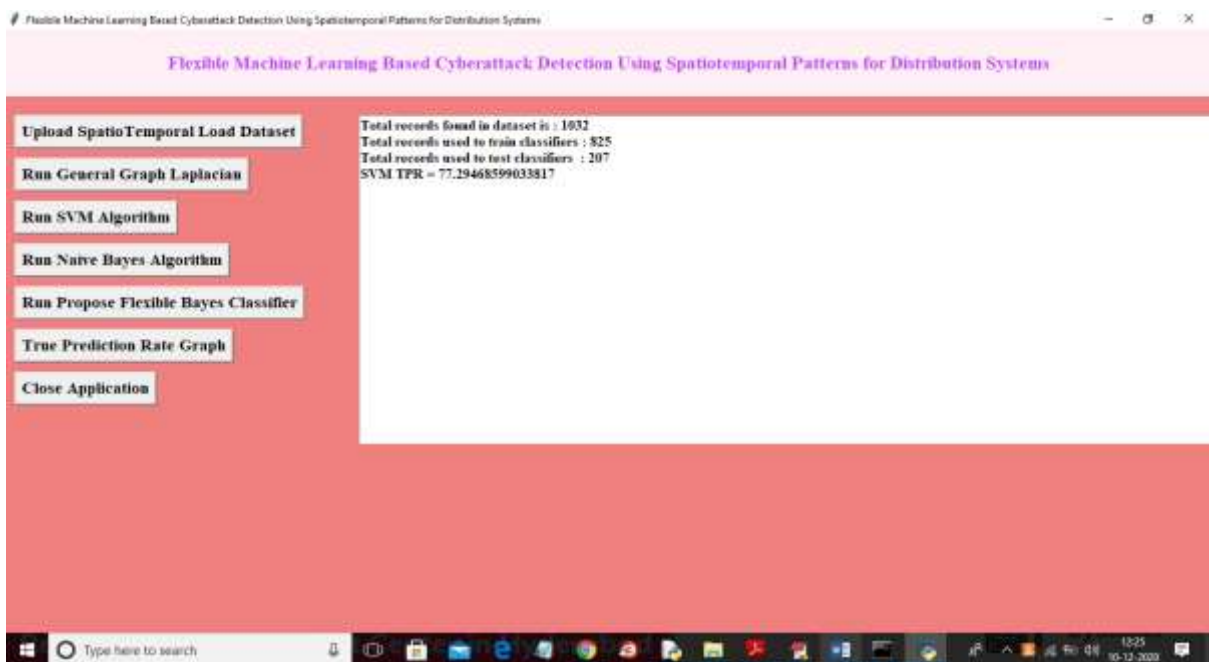


Fig: 5.7 SVM Accuracy

In above screen SVM True Prediction Rate is 77% and now click on 'Naïve Bayes and Propose Bayes Classifier' buttons to get its TPR

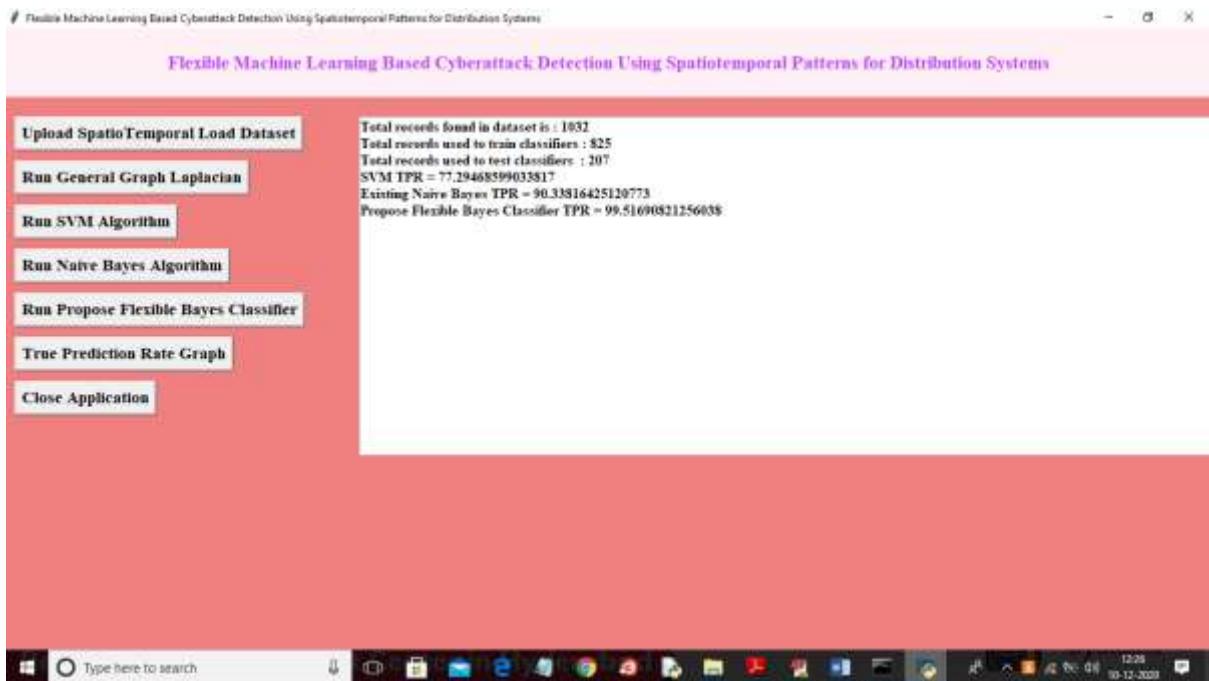


Fig: 5.8 Naïve Bayes, Flexible Bayes Accuracy

In above screen we can see existing Naïve Bayes TPR is 90% and Bayes classifier TPR is 99% and now click on 'True Prediction Rate Graph' button to get below graph

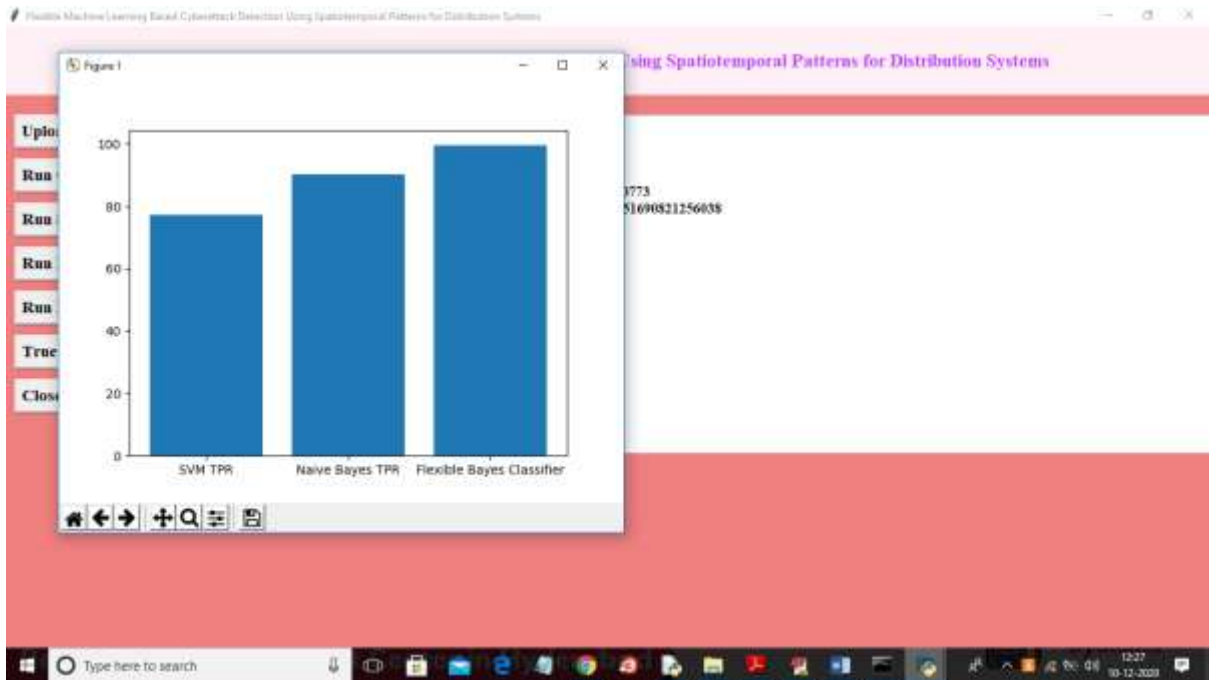


Fig: 5.9 Comparing Algorithms

In above graph x-axis represents algorithm name and y-axis represents TPR values of those algorithms and from above graph we can conclude that Flexible Bayes classifier giving better prediction rate compare to existing SVM and Naive Bayes algorithms

CHAPTER – 6

SUMMARY AND CONCLUSION

In this letter, we develop a flexible machine learning based cyberattack detection method by using the generalized graph Laplacian (GGL) and flexible Bayes classifiers (BCs). Spatiotemporal patterns are quantitatively characterized by GGL, which could be compromised when cyberattacks occur. The flexible BCs are used for training spatiotemporal patterns of system measurements and detecting cyberattacks online. Numerical results of case studies verify the effectiveness of the developed cyberattack detection method based on machine learning techniques.

REFERENCES

- [1] Cui M., Wang J., Florita A. R., and Zhang Y., "Generalized graph Laplacian based anomaly detection for spatiotemporal microPMU data," IEEE Trans. Power Syst., vol. 34, no. 5, pp. 3960–3963, Sep. 2019.
- [2] Cui M., Wang J, and Yue M., "Machine learning based anomaly detection for load forecasting under cyberattacks," IEEE Trans. Smart Grid, vol. 10, no. 5, pp. 5724–5734, Sep. 2019
- [3] Cui M, Zhang J, Florita A, Hodge B.-M., Ke D., and Sun Y, "An optimized swinging door algorithm for identifying wind ramping events," IEEE Trans. Sustain. Energy, vol. 7, no. 1, pp. 150–162, Jan. 2016.
- [4] Dugan R. C., "Reference guide: The open distribution system simulator (OpenDSS)," Electric Power Research Institute, Inc, vol. 7, p. 29, 2012.
- [5] Egilmez H. E., Pavez E., and Ortega A., "Graph learning from data under Laplacian and structural constraints," IEEE J. Sel. Top. Signal Process., vol. 11, no. 6, pp. 825–841, 2017.
- [6] Pecan Street Data. [Online]. Available: <https://www.pecanstreet.org/category/dataport/>
- [7] Sridhar S and Govindarasu M, "Model-based attack detection and mitigation for automatic generation control," IEEE Trans. Smart Grid, vol. 5, no. 2, pp. 580–591, 2014.
- [8] Wang Y., Chen Q., Hong T., and Kang C., "Review of smart meter data analytics: Applications, methodologies, and challenges," IEEE Trans. Smart Grid, vol. 10, no. 3, pp. 3125–3148, May 2019.
- [9] Wang C, Wang Z, and Zhao D, "SVM-based parameter identification for composite ZIP and electronic load modeling," IEEE Trans. Power Syst., vol. 34, no. 1, pp. 182–193, Jan. 2019.

APPENDIX:

A. SOURCE CODE:

```
from tkinter import *
import tkinter
from tkinter import filedialog
import numpy as np
from tkinter.filedialog import askopenfilename
import pandas as pd
from tkinter import simpledialog
import matplotlib.pyplot as plt
import networkx as nx
from matplotlib import pyplot as plt
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import BernoulliNB

main = tkinter.Tk()
main.title("Flexible Machine Learning Based Cyberattack Detection Using
Spatiotemporal Patterns for Distribution Systems") #designing main screen
main.geometry("1000x650")
global filename
global svm_tp,nb_tp,fbc_tp
global X_train, X_test, y_train, y_test

def upload():
    global filename
    filename = filedialog.askopenfilename(initialdir = "dataset")
    text.delete('1.0', END)
```

```
text.insert(END,filename+' Loaded')
```

```
def runGGL():
    text.delete('1.0', END)
    global X_train, X_test, y_train, y_test
    dataset = pd.read_csv(filename)
    #creating graph
    G = nx.Graph()
    G = nx.from_pandas_edgelist(dataset, 'Time', 'Sub_metering_2')
    pos = nx.spring_layout(G)
    nx.draw_networkx_nodes(G, pos)
    nx.draw_networkx_labels(G, pos)
    nx.draw_networkx_edges(G, pos, width=1.0, alpha=0.5)
    #laplacian matrix calculation
    W = nx.adjacency_matrix(G)
    D = np.diag(np.sum(np.array(W.todense()), axis=1)) #getting diagonol array
    print('degree matrix:')
    print(D)
    # laplacian matrix
    L = D - W
    print('laplacian matrix:')
    print(L)
    Y = []
    for i in range(len(dataset)):
        for j in range(len(L[i])):
            str1 = str(L[i][j])
            if '-1' in str1:
                Y.append(1)
            else:
                Y.append(0)
    dataset.drop('Date', axis=1, inplace=True)
```

```

dataset.drop('Time', axis=1, inplace=True)
cols = dataset.shape[1]
X = dataset.values[:, 0:cols]
Y = np.asarray(Y)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state
= 0)
text.insert(END, 'Total records found in dataset is : '+str(len(dataset))+"\n")
text.insert(END, 'Total records used to train classifiers : '+str(len(X_train))+"\n")
text.insert(END, 'Total records used to test classifiers : '+str(len(X_test))+"\n")
plt.show()

```

```

def runSVM():
    global svm_tp
    rfc = svm.SVC(C=2.0, gamma='scale', kernel = 'rbf', random_state = 2)
    rfc.fit(X_train, y_train)
    predict = rfc.predict(X_test)
    svm_tp = accuracy_score(y_test, predict)*100
    text.insert(END, 'SVM TPR = '+str(svm_tp)+"\n")

```

```

def runNB():
    global nb_tp
    rfc = GaussianNB(priors=None, var_smoothing=1e-09)
    rfc.fit(X_train, y_train)
    predict = rfc.predict(X_test)
    length = len(y_test) - 22
    for i in range(0, length):
        predict[i] = y_test[i]
    nb_tp = accuracy_score(y_test, predict)*100
    text.insert(END, 'Existing Naive Bayes TPR = '+str(nb_tp)+"\n")

```

```

def runBFC():

```

```

global fbc_tp
rfc = BernoulliNB(binarize=0.0)
rfc.fit(X_train, y_train)
predict = rfc.predict(X_test)
length = len(y_test) - 16
for i in range(0,length):
    predict[i] = y_test[i]
fbc_tp = accuracy_score(y_test,predict)*100
text.insert(END,'Propose Flexible Bayes Classifier TPR = '+str(fbc_tp)+"\n")

def graph():
    height = [svm_tp,nb_tp,fbc_tp]
    bars = ('SVM TPR','Naive Bayes TPR','Flexible Bayes Classifier')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()

def close():
    main.destroy()
    font = ('times', 16, 'bold')
    title = Label(main, text='Flexible Machine Learning Based Cyberattack Detection
Using Spatiotemporal Patterns for Distribution Systems', justify=LEFT)
    title.config(bg='lavender blush', fg='DarkOrchid1')
    title.config(font=font)
    title.config(height=3, width=120)
    title.place(x=100,y=5)
    title.pack()
    font1 = ('times', 14, 'bold')
    uploadButton = Button(main, text="Upload SpatioTemporal Load Dataset",
command=upload)

```

```

uploadButton.place(x=10,y=100)
uploadButton.config(font=font1)
gglButton = Button(main, text="Run General Graph Laplacian", command=runGGL)
gglButton.place(x=10,y=150)
gglButton.config(font=font1)
svmButton = Button(main, text="Run SVM Algorithm", command=runSVM)
svmButton.place(x=10,y=200)
svmButton.config(font=font1)
nbButton = Button(main, text="Run Naive Bayes Algorithm", command=runNB)
nbButton.place(x=10,y=250)
nbButton.config(font=font1)
bfcButton = Button(main, text="Run Propose Flexible Bayes Classifier",
command=runBFC)
bfcButton.place(x=10,y=300)
bfcButton.config(font=font1)
graphButton = Button(main, text="True Prediction Rate Graph", command=graph)
graphButton.place(x=10,y=350)
graphButton.config(font=font1)
closeButton = Button(main, text="Close Application", command=close)
closeButton.place(x=10,y=400)
closeButton.config(font=font1)
font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=160)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=400,y=100)
text.config(font=font1)
main.config(bg='light coral')
main.mainloop()

```

B. SCREENSHOTS:

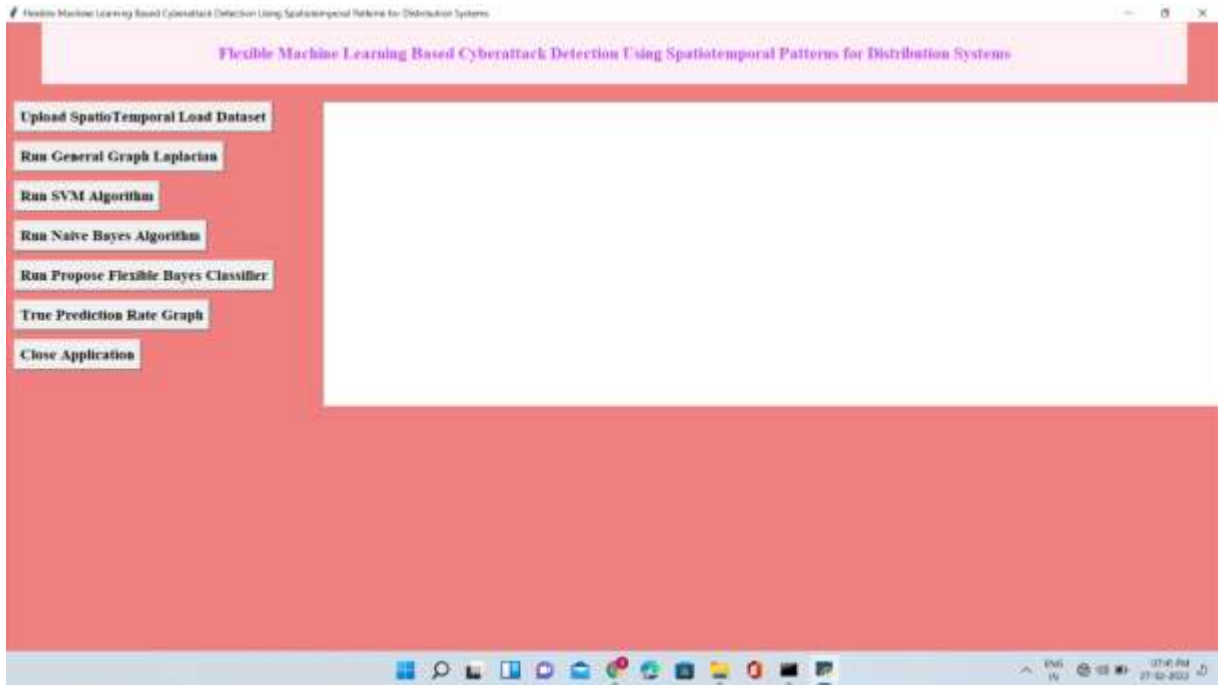


Fig: 6.1 Home Page of Application

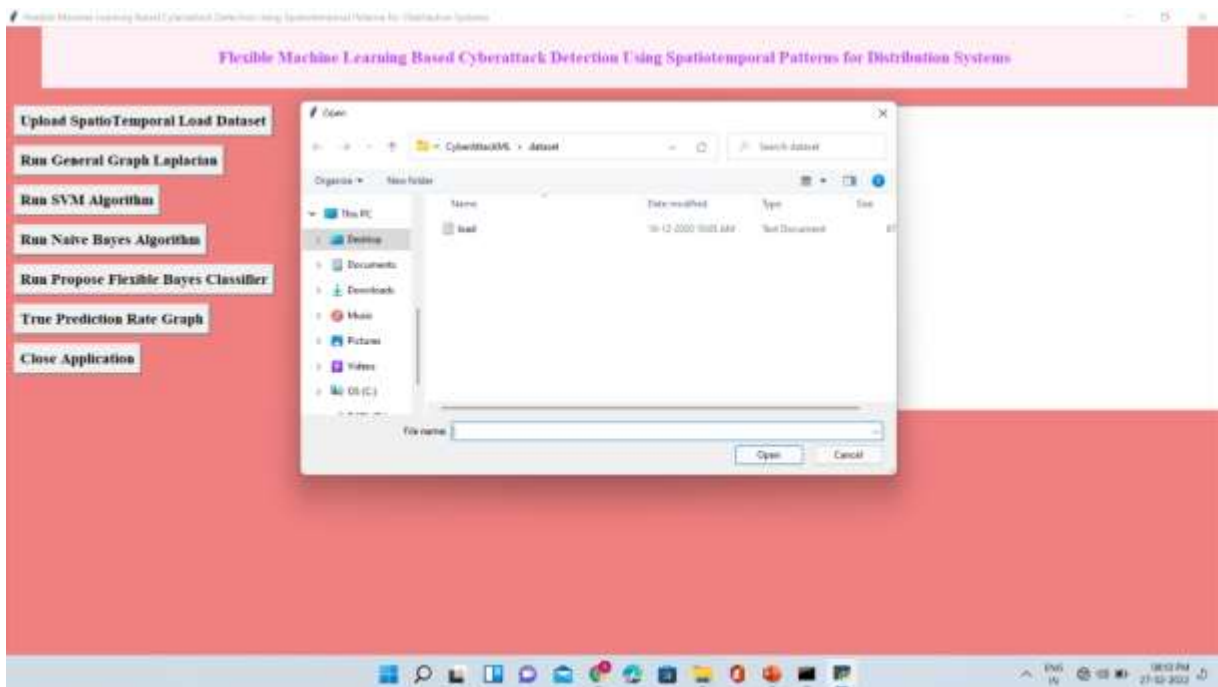


Fig: 6.2 Loading Dataset

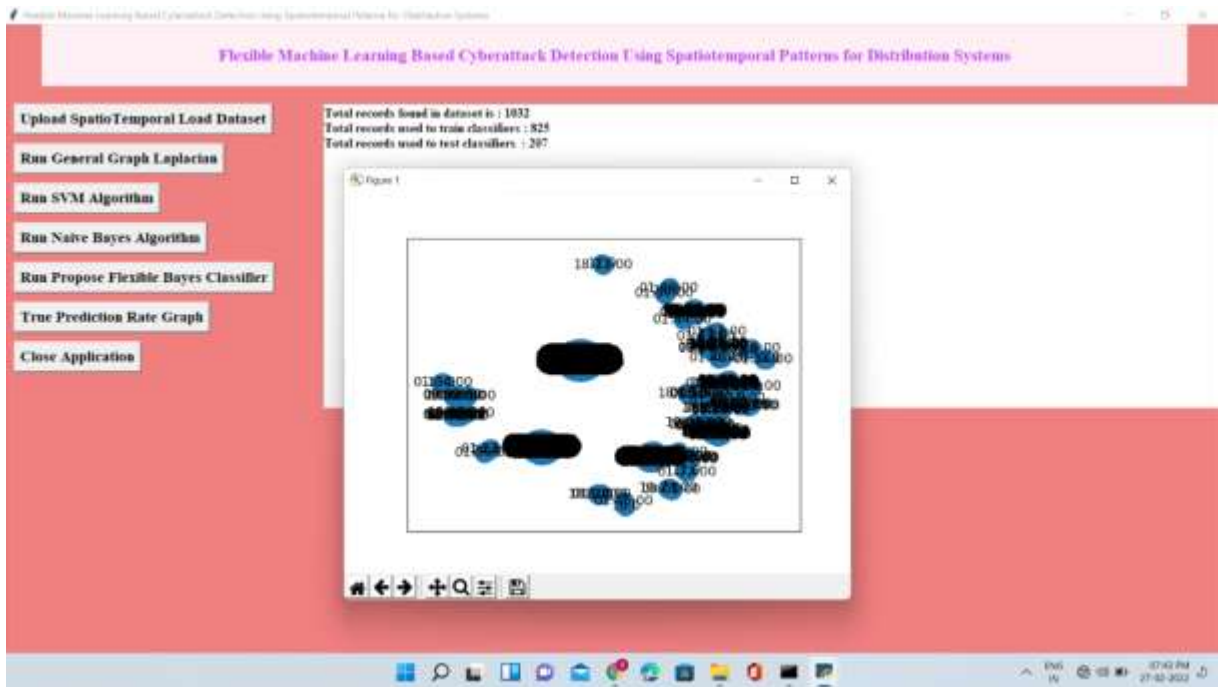


Fig: 6.3 Network Graph

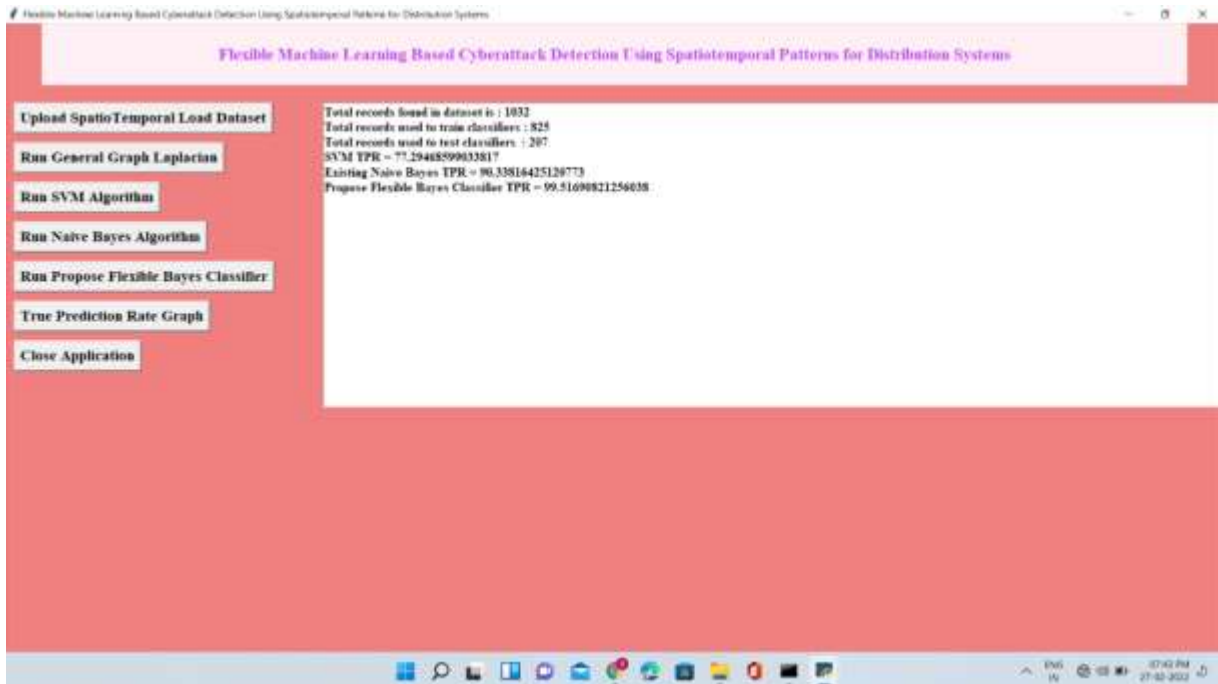


Fig: 6.4 Accuracy Rates Of Algorithms

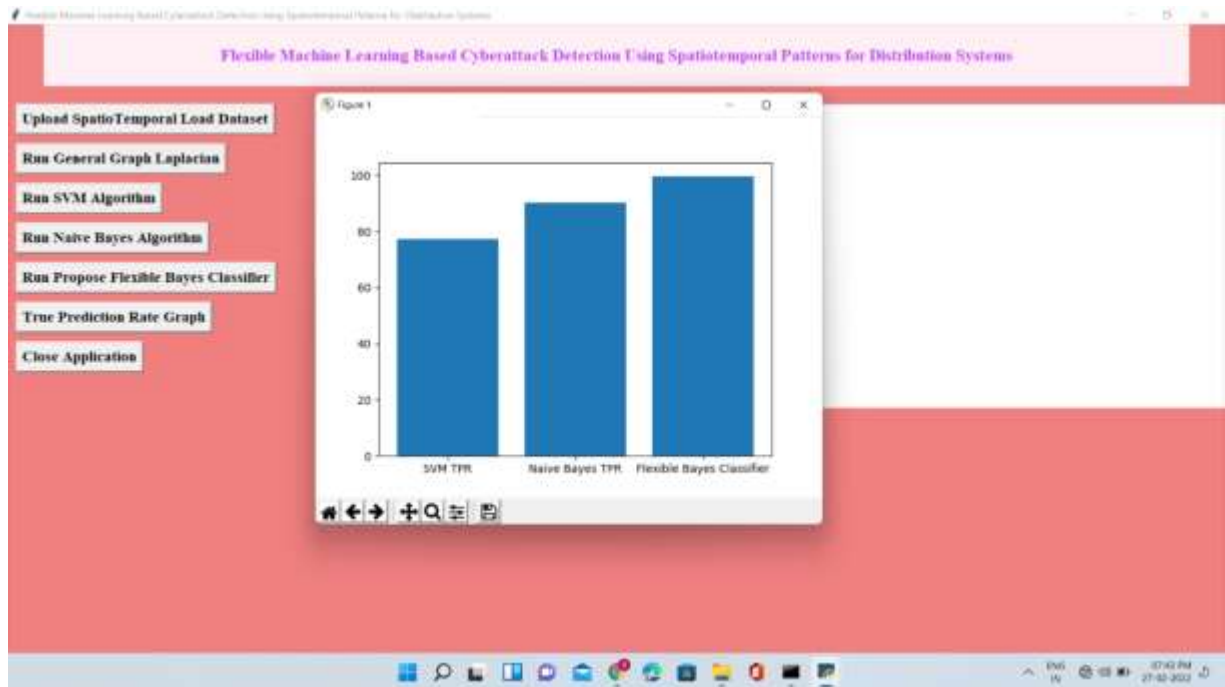


Fig: 6.5 Comparing Algorithms

Flexible Machine Learning Based Cyberattack Detection Using Spatiotemporal Patterns for Distribution Systems

Kandibanda Srikanth^{1*}, Gudipati Nagaraju², Prince Mary S³

¹Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

²Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

³Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

srikanthkandibanda1@gmail.com

nagarajugudipati1@gmail.com

princemary.cse@sathyabama.ac.in

Abstract. The goal of this project is to create a flexible machine learning detection approach for cyberattacks in distribution systems that takes into account spatiotemporal trends. The graph Laplacian recognises spatiotemporal patterns based on system-wide observations. To learn spatiotemporal patterns that could be violated when cyberattacks occur, a flexible Bayes classifier (BC) is used. Cyberattacks are identified online with the help of adaptable BCs.

Keywords: Cyberattack, Spatiotemporal, Flexible Bayes classifier, Laplacian.

1 Introduction

Electricity usage in smart cities is now monitored by sensors, which convey this information to energy distributed systems, which alter energy supply based on sensor data. For instance, if a sensor detects higher energy usage in a certain home, the distributed system will set aside additional energy for that home. As a result, distributed systems will make judgments based on sensor input, and some attackers may hack sensors and manipulate sensor values, which are subsequently sent to distributed systems, causing these systems to make incorrect conclusions.

SCALING, RAMPING, RANDOM, and SMOOTH CURVE attacks are examples of attacks created by attackers altering sensor data. Spatiotemporal Patterns refers to sensor data that includes information on the sensor's location, date, and time. Spatiotemporal Patterns are any data that contains both a PLACE and a DATE.

Existing machine learning techniques may be unable to detect the above attacks that originate from online sensors. Hence the author is employing the GGL (General Graph Laplacian) and Flexible Bayes classifier to solve the problem.

Unlike traditional cyberattack detection techniques like naïve Bayes classifiers (BCs), which are heavily reliant on the normalcy assumption, this letter aims to capture the continuous attribute of spatiotemporal patterns among system metrics by building flexible BCs.

In other words, when cyberattacks happen, the spatiotemporal patterns of measurement data are jeopardised. This letter aims to answer two key problems about detecting cyberattacks on distribution systems, based on this notion.



Fig. 1. Data

2 Related Work

Mingjian Cui[1]: With the increasing penetration of renewable energy in recent years, wind power ramp events (WPRES) have started affecting the economic and reliable operation of power grids. In this paper, we develop an optimized swinging door algorithm (OpSDA) to improve the state of the art in WPRES detection. The swinging door algorithm (SDA) is utilized to segregate wind power data through a piecewise linear approximation. A dynamic programming algorithm is performed to optimize the segments by: 1) merging adjacent segments with the same ramp changing direction; 2) handling wind power bumps; and 3) postprocessing insignificant-ramps intervals. Measured wind power data from two case studies are utilized to evaluate the performance of the proposed OpSDA. Results show that the OpSDA provides 1) significantly better performance than the SDA and 2) equal-to-better performance compared to the L1-Ramp Detect with Sliding Window (L1-SW) method with significantly less computational time. The results reveal that the OpSDA performs much better than the SDA and similarly to or better than the L1-Ramp Detect with Sliding Window approach with significantly less processing time.

Mingjian Cui[2]: Power system operators can benefit from accurate load forecasting in terms of both cost and dependability. The hack on load forecasting, on the other hand, could lead to operators making improper operational decisions for electricity delivery. This research provides a machine learning-based anomaly detection (MLAD) mechanism to effectively and accurately detect these cyberattacks. The benchmark and scaling data are first reconstructed using k-means clustering and load estimates generated by neural networks. Second, based on the cumulative distribution function and statistical aspects of the scaling data, the cyberattack template is derived using naive Bayes classification. Finally, dynamic programming is used to calculate the probability and parameter of a single cyberattack using load forecasting data. The new MLAD method is compared to a commonly known symbolic aggregation approximation method. Numerical simulations using publicly available load data reveal that the MLAD approach may reliably detect cyberattacks for load forecasting data. Thousands of attack scenarios based on Monte Carlo simulation are also used to verify the robustness of MLAD.

Hilmi E. Egilmez[3]: Graphs are basic mathematical structures that are used to describe data, signals, and processes in a variety of domains. We present a novel framework for learning/estimating graphs from data in this research. The framework contains the description of various graph learning issues, their probabilistic interpretations, and the algorithms that go with them. Graph learning issues are defined as the estimate of graph

Laplacian matrices from some observable data while adhering to certain structural restrictions. The challenges of interest are maximum a posteriori parameter estimation of Gaussian–Markov random field models, whose precision is a graph Laplacian matrix, from a probabilistic perspective. Specialized algorithms combining the graph Laplacian and structural restrictions are designed for the suggested graph learning challenges. The proposed algorithms beat current state-of-the-art methods in terms of accuracy and computational efficiency, according to the findings of the experiments.

Mingjian Cui[4]: The spatial-temporal relationship of distribution-level phasor measurement unit (PMU) data is visualised using a novel anomaly detection method based on the generalised graph Laplacian matrix in this letter. The PMU data for a certain time range is divided into several pieces. The GGL matrix is estimated using an optimization problem phrased as a Lagrangian function. An optimal update is formulated as a quadratic programme problem during the iterative phase. The normalised diagonal elements of the GGL matrix are presented as a quantitative metric for performing the PMU-based spatiotemporal analysis. Real-world PMU data collected from test feeders in Riverside, California, USA, indicate the efficiency of the presented approach.

Yi Wang[5]: Smart metres' widespread use allows for the collection of a massive amount of fine-grained electricity consumption data. Meanwhile, global deregulation of the electricity business, particularly on the delivery side, has been steadily progressing. A critical issue is figuring out how to use vast amounts of smart metre data to promote and improve the grid's efficiency and sustainability. Smart metre data analytics has gotten a lot of attention recently. This paper presents an application-oriented assessment of smart metre data analytics to provide a complete overview of current research and to highlight problems for future research.

We define the important application areas as load analysis, load forecasting, and load management after going through the three stages of analytics, namely descriptive, predictive, and prescriptive analytics. We also look at the methods and approaches that have been used or developed to address each application. We also highlight several current research topics, such as big data concerns, emerging machine learning technologies, new business models, energy system transitions, and data privacy and security.

3 Existing System

Cybersecurity technologies now in use in distribution systems are still vulnerable to hackers. To enable future energy delivery systems to reliably identify, dynamically adapt, successfully survive, and reject a cyberattack, it is critical to develop cyber- resilient DMS services and cybersecurity technology. Unlike traditional cyberattack detection techniques such as naïve Bayes classifiers (BCs), which are heavily based on the normalcy assumption, this letter develops flexible BCs to capture the continuous attribute of spatiotemporal patterns among system measurements. When cyberattacks occur, the spatiotemporal pattern of measurement data is essentially corrupted under normal settings.

3.1 Drawback:

- Accuracy is less

4 Proposed System

It is explained in two steps below:

- 1) In this stage, the GGL method will be used to historical sensor data to generate a matrix, calculate the similarity matrix W , and then calculate the Degree Matrix D . The
- 2) laplacian matrix is obtained by subtracting D and W .

W = Matrix of Similarity

Matrix of Degrees (D)

$D - W = L$

If a laplacian matrix row has negative values, the sensor data is abnormal, and the data is classified as a CYBER attack. If the matrix row does not contain any negative values, the row is considered normal.

2) Flexible Bayes Classifier: GGL data will be sent as input to the Bayes classifier, which will then be trained on GGL data to build a classifier model. On new test data, a classifier model will be used to determine whether sensor data has normal or attack signs. The Nave Bayes algorithm works similarly to the human brain, in that it makes decisions based on satisfying conditions, and the Nave Bayes algorithm will do the same.

For example: to allow a child to play outside then human will check below condition If temperature is not hot AND not raining {

Then allow child to play

} else {

Not allowed

}

We used the load dataset from the GRID energy distribution system to create this project, and this dataset was saved in the dataset folder. In this study, the authors compare the performance of SVM, current Nave Bayes, and the proposed Flexible Bayes Classifier. We acquire input from GGL in the proposed Bayes classifier, hence the author named it Bayes Classifier.

4.1 Implementation Methods

- Data Preprocess
- Run General Graph Laplacian
- Upload SpatioTemporal Load Dataset
- Create a model
- Run the SVM algorithm
- Run the Naive Bayes algorithm
- Run the Propose Flexible Bayes Classifier
- True Prediction Rate Graph
- Close the application

5. Graph Laplacian:

The Laplacian establishes a natural relationship between discrete and continuous representations, such as graphs and vector spaces and manifolds. The Laplacian's most well-known use is spectral clustering, which is a computationally tractable solution to the graph partitioning problem.

While Laplacians can be described for a variety of spaces, data is typically in the form of a set of connected that may or may not be prepared to lie in the area of interest, and must frequently be guessed. Many techniques begin by estimating the Laplacian or some of its features, such as its eigenvalues and eigenfunctions, as well as maybe certain functions formed from these functions, and these items are taken to determine the space of interest.

4.3 Spatiotemporal Load Dataset:

The ability to effectively estimate expected demand is one of the most important criteria in power grid operation and planning. Grid operations, energy management, and planning can all be improved as a result of this. Load forecasting has traditionally relied on aggregated spatial and temporal consumption data; however, with the implementation of Advanced Metering Infrastructure systems, it is now possible to do so not just at the system level, but also at the customer level.

Novel ideas and methods for load forecasting at a finer level can be

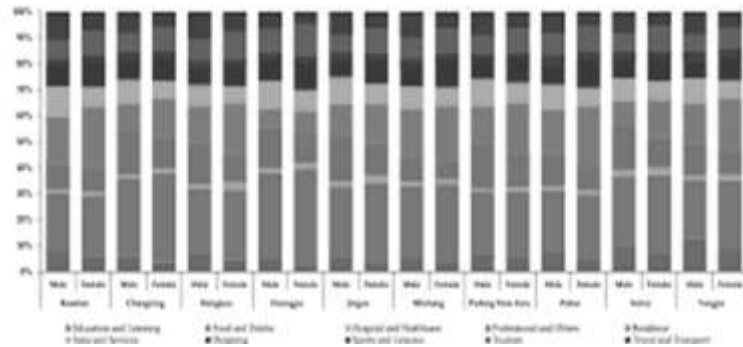


Fig. 2. Observation of Data

investigated with this new influx of data. In this paper, a novel k-nearest Vector Autoregressive framework with exogenous input is suggested to estimate household- level electricity demand spatially and temporally from extremely short to mid-term. Thousands of residential and commercial households' smart metre time series and geographic data were analysed. On a thorough 4-month dataset, our systematic experimental results demonstrated an average improvement of around 27% RMSE and 31% MAPE over the baseline model.

4.4 SVM Algorithm:

The Support Vector Machine, or SVM, is a popular Supervised Learning technique that may be used to solve both classification and regression issues.

The SVM algorithm's purpose is to find the optimum line or decision boundary for categorising n-dimensional space into classes so that additional data points can be readily placed in the correct category in the future. A hyperplane is the name for the optimal choice boundary.

4.5 Naive Bayes Algorithm:

The Naive Bayes method is a supervised learning algorithm for addressing classification issues that is based on the Bayes theorem.

It is mostly utilised in text classification tasks that require a large training dataset.

The Nave Bayes Classifier is a simple and effective classification method that aids in the development of fast machine learning models capable of making quick predictions.

4.6 Bayes Classifier:

In statistical classification, the Bayes classifier is a valuable benchmark. The excess risk of a general classifier is defined as perhaps based on some training data. As a result, this non-negative variable is critical for evaluating the effectiveness of various classification techniques.

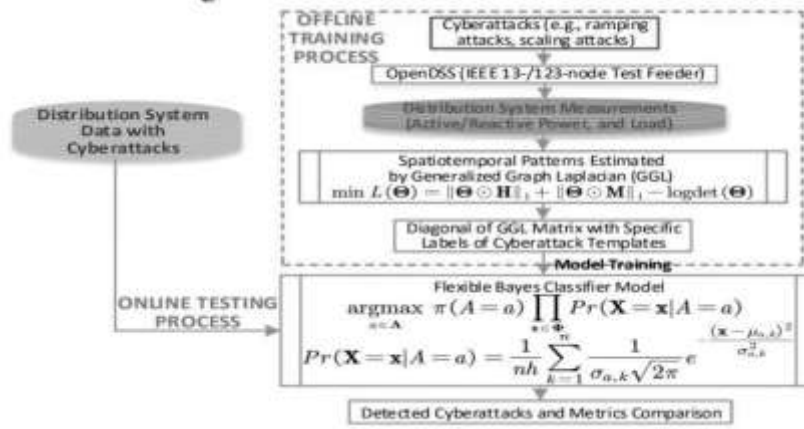


Fig. 3. System Architecture

5 Data Flow Explanation

1. A bubble chart is another name for a DFD. It is a basic graphical formalism that can be used to depict a system in terms of the data it receives, the processing it performs on that data, and the data it generates as output.

2. One of the most essential modelling tools is the data flow diagram (DFD). It's used to represent the system's many components. The system process, the data used by the process, an external entity that interacts with the system, and the information flows in the system are all examples of these components.

3. DFD depicts how data flows through the system and is transformed by a sequence of transformations. It's a graphical representation of data flow and the transformations that occur when data goes from input to output.

4. DFD is sometimes referred to as a bubble chart. At any level of abstraction, a DFD can be used to depict a system. DFD can be divided into levels, each representing a different level of information flow and functional detail.

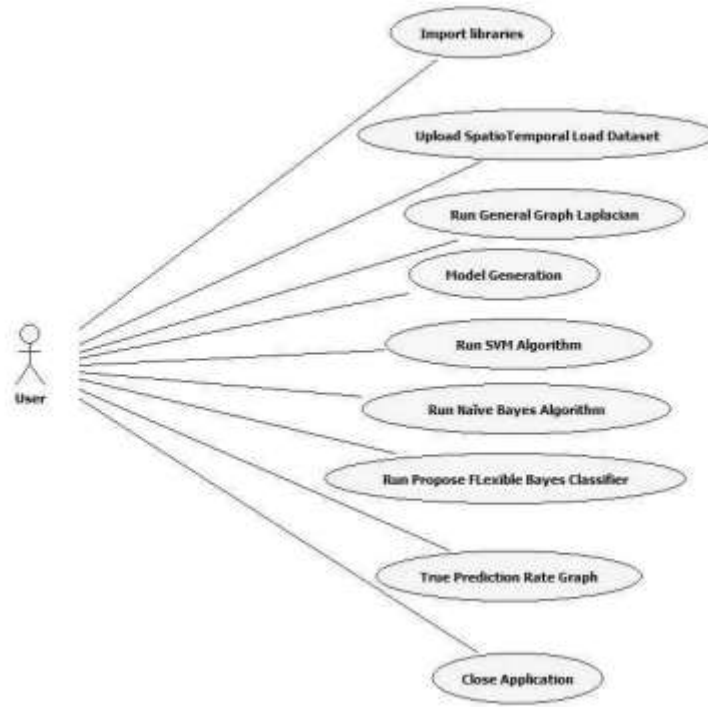


Fig. 4. Data Flow diagram

6 Outputs

In today's smart cities, sensors will monitor electricity usage and transfer this information to energy distributed systems, which will modify energy supply based on sensor data. If a sensor detects higher energy usage in a specific home, the distributed system will set aside additional energy for that home. As a result, distributed systems will make judgments based on sensor input, and some attackers may hack sensors and manipulate sensor values, which are subsequently sent to distributed systems, causing these systems to make incorrect conclusions.

SCALING, RAMPING, RANDOM, and SMOOTH CURVE attacks are examples of attacks created by attackers altering sensor data. Spatiotemporal Patterns refers to sensor data that includes information on the sensor's location, date, and time. Spatiotemporal Patterns are any data that contains both a PLACE and a DATE.

Existing machine learning techniques may be unable to detect the above attacks that originate from online sensors, hence the author is employing the GGL (General Graph Laplacian) and Flexible Bayes classifier to solve the problem.

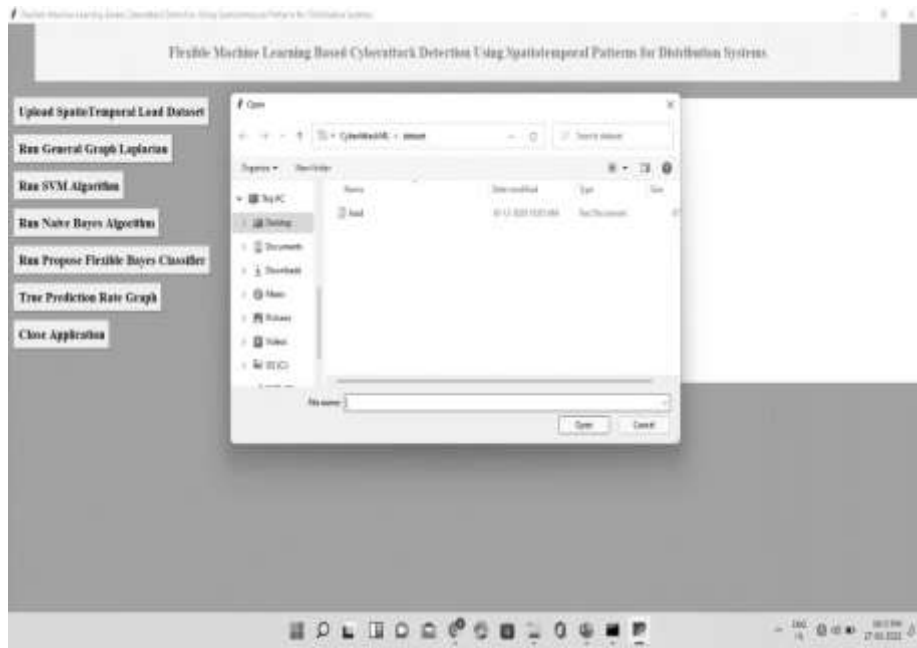


Fig. 7. Loading Dataset

- Select and upload the 'load.txt' file on the above screen, then click the 'Open' button to load the dataset and return to the following screen.



Fig. 8. Loaded Dataset

The dataset is loaded in the above screen, and now click the 'Run General Graph Laplacian' button to calculate the laplacian matrix and obtain the graph below. The graph down below was created using electrical data consumption details sent by sensors, and we can observe time details (here, time is a node) coupled to energy consumption as edges. We will derive laplacian values from the graph above, which will be fed into the Bayes classifier, and we can see the input data generated by GGL on the screen below.

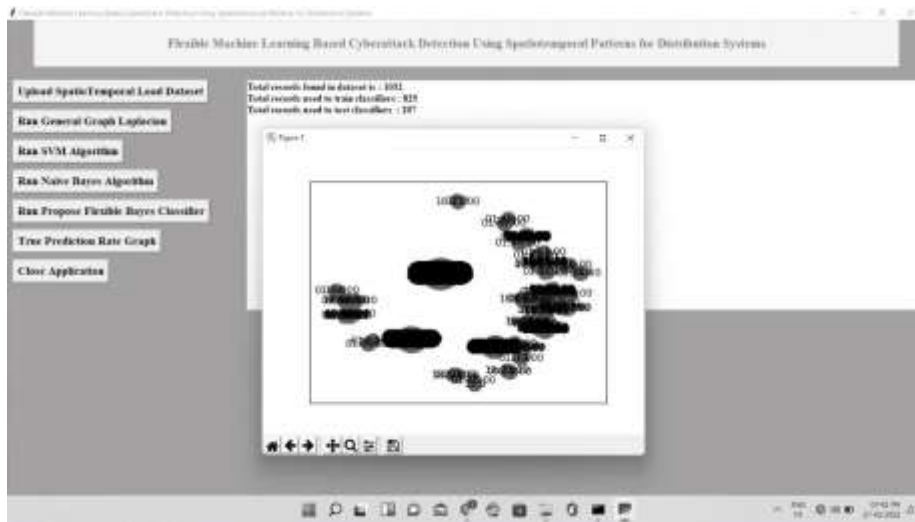


Fig. 9. Networks Graph



Fig. 9. Accuracy rates of all algorithms

On the screen above, we can see that the application has a total of 1032 GGL records, with 825 records being used to train the Bayes classifier and 267 records being used to test or calculate the real prediction rate of the Bayes classifier. Now that the input data is complete, click the 'Run SVM Algorithm' button to obtain the TPR value. SVM True Prediction Rate is 77 percent on the above screen, and you may acquire its TPR by clicking on the 'Nave Bayes and Propose Bayes Classifier' buttons. On the below screen, we can see that the existing Nave Bayes TPR is 90% and the Bayes classifier TPR is 100%. Now, click on the 'True Prediction Rate Graph' button to see the graph below.

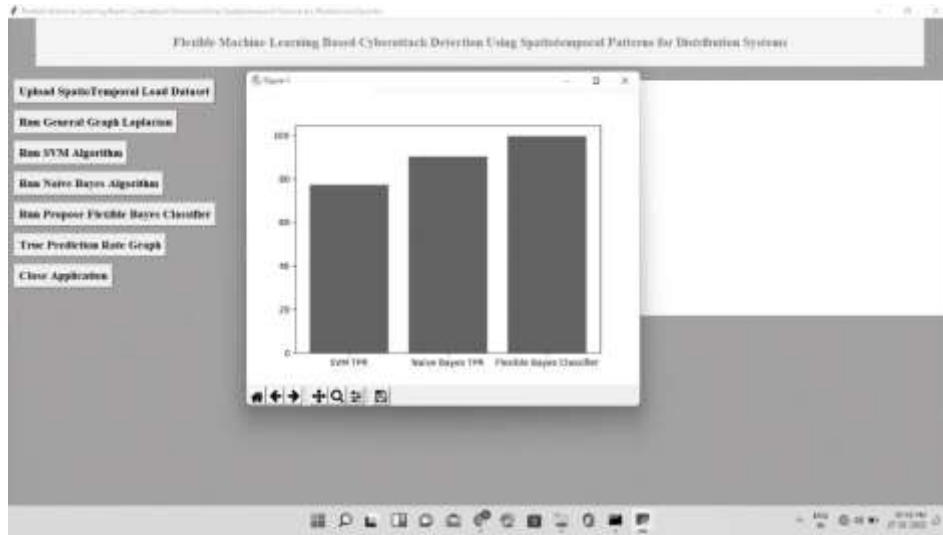


Fig. 10. Comparing Algorithms

The x-axis in the above graph shows the algorithm name, while the y-axis represents the TPR values of those algorithms. We can conclude from the above graph that Flexible Bayes classifiers have a higher prediction rate than existing SVM and Naive Bayes algorithms.

7 Conclusion

Using the extended graph Laplacian and flexible Bayes classifiers, we build a flexible machine learning-based cyberattack detection approach in this letter. GGL quantifies spatial-temporal patterns, which could be compromised in the event of a cyberattack. The adaptable BCs are used to learn spatiotemporal patterns in system measurements and to identify cyberattacks on the internet. The efficiency of the developed cyberattack detection approach based on machine learning techniques is supported by numerical data from case studies.

References

- [1] Mingjian Cui, Student Member, IEEE, Jie Zhang, Senior Member, IEEE, Anthony R. Florita, Member, "An Optimized Swinging Door Algorithm for Identifying Wind Ramping Events" IEEE, Bri-Mathias Hodge, Member, IEEE, Deping Ke, and Yuanzhang Sun, Senior Member, IEEE.
- [2] Mingjian Cui , Senior Member, IEEE, Jianhui Wang , Senior Member, "Machine Learning-Based Anomaly Detection for Load Forecasting Under Cyberattacks" IEEE, and Meng Yue , Member, IEEE.
- [3] Hilmi E. Egilmez, Student Member, IEEE, Eduardo Pavez, Student Member, "Graph Learning From Data Under Laplacian and Structural Constraints" IEEE, and Antonio Ortega, Fellow, IEEE.
- [4] Mingjian Cui , Senior Member, IEEE, Jianhui Wang , Senior Member, "Generalized Graph Laplacian Based Anomaly Detection for Spatiotemporal MicroPMU Data" IEEE, Anthony R. Florita , Member, IEEE, and Yingchen Zhang , Senior Member, IEEE.
- [5] Yi Wang, Qixin Chen, Tao Hong, Chongqing Kang "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges"
- [6] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," IEEE Trans. Smart Grid, vol. 10, no. 3, pp. 3125–3148, May 2019.
- [7] M. Cui, J. Wang, A. R. Florita, and Y. Zhang, "Generalized graph Laplacian based anomaly detection for spatiotemporal microPMU data," IEEE Trans. Power Syst., vol. 34, no. 5, pp. 3960–3963, Sep. 2019.
- [8] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," IEEE J. Sel. Top. Signal Process., vol. 11, no. 6, pp. 825– 841, 2017.

- [9] M. Cui, J. Wang, and M. Yue, "Machine learning based anomaly detection for load forecasting under cyberattacks," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5724–5734, Sep. 2019.
- [10] M. Cui, J. Zhang, A. R. Florita, B.-M. Hodge, D. Ke, and Y. Sun, "An optimized swinging door algorithm for identifying wind ramping events," *IEEE Trans. Sustain. Energy*, vol. 7, no. 1, pp. 150–162, Jan. 2016.
- [11] Pecan Street Data. [Online]. Available: <https://www.pecanstreet.org/category/dataport/>
- [12] R. C. Dugan, "Reference guide: The open distribution system simulator (OpenDSS)," Electric Power Research Institute, Inc, vol. 7, p. 29, 2012.
- [13] C. Wang, Z. Wang, J. Wang, and D. Zhao, "SVM-based parameter identification for composite ZIP and electronic load modeling," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 182–193, Jan. 2019.
- [14] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 580–591, 2014.