Intrusion Detection in Internet of Things Systems: A Feature Extraction with Naive Bayes Classifier Approach

[1]Juan Carlos Juarez Vargas, [2,3]Hayder M A Ghanimi, [4]Sivaprakash S, [5]Amarendra M, [6]Rajendiran M and [7]Sheylla L Cotrado Lupo

[1]Faculty of Statistical and Computer Engineering, Universidad Nacional del Altiplano de Puno, P.O. Box 291, Puno, Peru.
[2]Biomedical Engineering Department, College of Engineering, University of Warith Al-Anbiyaa, Karbala, Iraq.
[3]Computer Science Department, College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq.
[4]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, TamilNadu, India.
[5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, 522502, Andhra Pradesh, India.
[6]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, 600123, TamilNadu, India.
[7]Faculty of Economic and Accounting Administrative Sciences, Universidad Andina del Cusco, 080104 Cusco Perú.
[1]jjuarezv@unap.edu.pe, [2,3]hayder.alghanami@uowa.edu.iq, [4]sivaprakash.s@vit.ac.in, [5]amarendra@kluniversity.in, [6]mrajendiran@panimalar.ac.in, [7]sheyllacotradolupo@gmail.com

Correspondence should be addressed to Sivaprakash S : sivaprakash.s@vit.ac.in.

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404003
Received 10 April 2023; Revised from 18 August 2023; Accepted 25 September 2023.
Available online 05 January 2024.

Abstract – The Internet of Things (IoT) has proliferated, transitioning from modest home automation to encompass sectors like healthcare, agriculture, transportation, and manufacturing. This evolution is characterized by devices' ability to autonomously gather, disseminate, and analyze data, leading to improved real-time decision-making, predictive insights, and customized user experiences. The ubiquity of IoT, while promising, introduces significant data security concerns. The vast number of interlinked devices and diverse and often insufficient security features make them vulnerable to cyber threats, emphasizing the need for robust security mechanisms. Intrusion Detection Systems (IDS) have traditionally acted as vital guards against such threats; however, with the ever-increasing data in the IoT, traditional IDS models, such as Naive Bayes, face processing speed and accuracy challenges. This paper introduces a novel model, "FE+NB," which merges advanced Feature Extraction (FE) with the Naive Bayes (NB) classifier. Central to this model is the "Temporal-Structural Synthesis" technique tailored for IoT traffic data, focusing on data compression, temporal and structural analyses, and Feature Selection (FS) using mutual information. Consequently, the model enhances efficiency and accuracy in Intrusion Detection (ID) in complex IoT networks.

Keywords – Intrusion Detection, Feature Extraction, Naïve Bayes, Internet of Things, Accuracy.

I. INTRODUCTION

The evolution of the Internet of Things (IoT) has marked a transformative phase in the digital era, demonstrating unparalleled growth over recent years. Originating from mere home automation concepts, IoT has now branched out, permeating many sectors, including healthcare, agriculture, transportation, and manufacturing. The ability of devices to collect, share, and interpret data without human intervention has redefined operational efficiencies, enabling real-time decision-making, predictive analytics, and personalized user experiences [1]. From wearable health monitors and smart thermostats to interconnected industrial machines and smart city infrastructures, the footprint of IoT is vast, symbolizing a future where almost every device is interlinked, enhancing the overall quality of life and operational efficiency across fields.

While the expansive growth of IoT presents numerous opportunities, it simultaneously contains plenty of challenges, particularly concerning data security [2]. With billions of devices interconnected, each transmitting and receiving data, the vulnerability to cyber threats has significantly heightened. These devices, often with limited security features, become prime targets for cyber-attacks, leading to data breaches, unauthorized access, and even large-scale network disruptions [3]. Furthermore, the vast heterogeneity of IoT devices and inconsistent security standards complicate the development of uniform protective measures [4]. Given that these threats can compromise personal privacy and enterprise-level information, there is an imperative need for robust security frameworks. Among the many protective solutions, Intrusion Detection Systems (IDS) emerge as pivotal, serving as the first line of defense by monitoring and flagging suspicious activities in real-time, ensuring the integrity of IoT ecosystems [5]. **Fig 1** shows an architecture of Machine Learning (ML) and Deep Learning (DL)-based IDS.

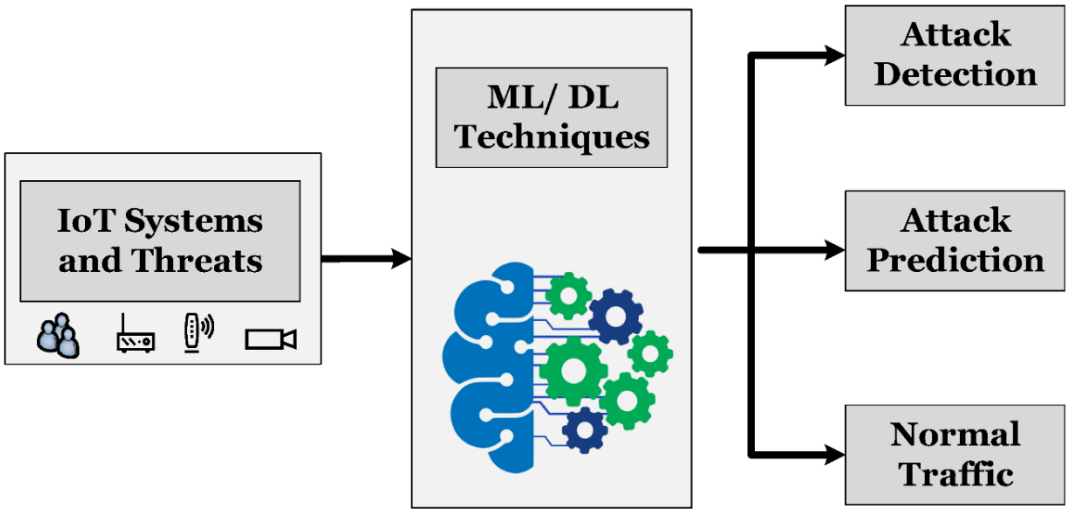

Fig 1. ML/DL-based IDS

IDS have long stood as sentinels against cyber threats, leveraging various algorithms and techniques to monitor and identify suspicious activities. Among these techniques, Naive Bayes (NB) has garnered attention for its probabilistic approach, which utilizes Bayes' theorem to calculate the likelihood of an event based on prior knowledge of related conditions [6]. While its simplicity and computational efficiency make it suitable for real-time analysis, the burgeoning data volume in IoT systems poses a challenge. As the IoT landscape continues to proliferate, the sheer magnitude of data generated by countless devices can overwhelm traditional IDS models, including NB, leading to slower response times and potential misclassifications [7]. Hence, to ensure the optimal performance of these models, addressing this data volume becomes critical. One strategic approach to manage this challenge is Feature Extraction (FE), which reduces vast datasets into their most salient attributes, thus reducing dimensionality and enhancing the efficiency of IDS [8]. Properly executed FE not only streamlines the data processing but also amplifies the accuracy and reliability of the IDS.

In light of the challenges existing IDS faces, particularly the struggle with the vast volumes of IoT data, a pressing need emerges for an advanced, refined approach. It is evident that while methods like NB have proven effective in many contexts, their performance can be significantly hampered without proper data management. Addressing this precise challenge, the proposed model, dubbed "FE+NB," integrates advanced FE techniques with the NB classifier to revolutionize intrusion detection in IoT systems. The crux of this model lies in its unique "Temporal-Structural Synthesis" approach to FE, which is meticulously tailored for IoT traffic data. This innovative method commences with data compression using autoencoders, transitioning through intricate temporal and structural analyses, and culminates with a rigorous Feature Selection (FS) based on mutual information metrics. As a result, the model refines the raw data into its most significant attributes, reducing redundancy and noise. When these meticulously extracted features are input into the NB classifier, the system achieves superior efficiency, precision, and speed, outperforming many traditional methods. Through this constructive interaction of sophisticated FE and the probabilistic strengths of NB, the FE+NB model promises a robust and scalable solution for IDS in increasingly complex IoT networks.

The paper is organized as follows: Section 2 presents the literature study, Section 3 presents the materials and methods used in this work, Section 4 presents the results, and Section 5 presents the conclusion of the work.

II. LITERATURE SURVEY

The domain of IDS for the IoT has garnered substantial research interest over recent years. A significant contribution comes from [9], who proposed a two-phase IDS focusing on different data types and deploying varied versions of the NB classifier for precise categorization, later using an elliptic envelope to classify benign data further. The method achieved impressive accuracy across multiple standard datasets, including NSL-KDD and CIC-IDS2017. This emphasis on data categorization and classification is mirrored by [10], who introduced a clustering-based classification technique using sophisticated algorithms like Anticipated Distance-based Clustering (ADC) in conjunction with Density-Based Spatial clustering of applications with noise (DBScan). The method utilized the Likelihood Naïve Bayes (LNB) for classification after clustering. Results demonstrated the superiority of the ADC-DBScan-LNB model in detecting intrusions.

Furthering the exploration into FS and classification, [11-13] leveraged deep FE and wrapper-based FS techniques, focusing on algorithms such as NB, to optimize the IDS.

Their approach reported high detection accuracy on the Aegean Wi-Fi Intrusion Dataset. Other notable models, like that by [14-18], introduced the Unified IDS for IoT, specifically tailored to handle a variety of attacks using the UNSW-NB15 dataset.

Simultaneously, [19-21] devised an M-IDM architecture that used real data from healthcare IoT devices to categorize intrusions. In an application-focused study, [22-23] developed a machine learning-based system to identify IoT network attacks, using real-world data from sensor networks and applying classifiers like Naïve Bayes for accurate intrusion classification. Collectively, these studies illuminate the evolution and potential of NB-centered approaches in IoT-IDS [24-25].

III. PROPOSED MODEL

Problem Definition

Intrusion detection in loT systems entails classifying network activities as 'Normal' or 'Malicious'. We define this problem mathematically as follows:

Variables

- Let x be a feature vector representing a loT network activity, where $x = (x_1, x_2, \dots, x_n)$. Each x_i represents a specific FE from the network traffic, such as packet size transmission rate.
- Let C be a random variable representing the class of the network activity. C can take two values:
- $C = 0$ representing 'normal' activity.
- $C = 1$ representing 'malicious' activity.

Objective

Determine the posterior probability $P(C \mid x)$ that a given network activity with feature vector x belongs to class C.

Using the Bayes Theorem:

$$P(C \mid x) = \frac{P(x|C)P(C)}{P(x)} \tag{1}$$

where:

- $P(C \mid x)$ is the posterior probability.
- $P(x \mid C)$ is the likelihood, which is the probability of observing feature vector x given class C.
- $P(C)$ is the prior probability of class C, representing this work's initial belief before observing x.
- $P(x)$ is the evidence, which can be computed as $P(x) = P(x \mid C = 0)P(C = 0) + P(x \mid C = 1)P(C = 1)$.

To classify a loT network activity, we would compute $P(C = 0 \mid x)$ and $P(C = 1 \mid x)$ and assign x to the class with the higher probability, EQU (2)

$$\begin{cases} 0 & \text{if } P(C = 0 \mid x) > P(C = 1 \mid x) \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

Problem Statement

Given a training dataset $D = \{(x^{(1)}, C^{(1)}), (x^{(2)}, C^{(2)}), \dots, (x^{(m)}, C^{(m)})\}$ of m labelled network activities, train a NB classifier to approximate the likelihoods $P(x_i \mid C)$ for each feature x_i and prior probabilities $P(C)$ to effectively classify new unseen loT network activities.

Dataset

The research relies on publicly available datasets to study intrusion detection in IoT systems. Selected sources include the CICIDS2017 and NSL-KDD. These datasets provide a range of traffic from benign interactions to various attack vectors. FE from these datasets focuses on network packet attributes: source and destination IP addresses, ports, protocol types, and

timestamps. Additionally, statistical features such as mean packet size, standard deviation of packet size, packet rate, and byte rate are computed. The datasets also offer content details, like flags and payload lengths, and temporal metrics, such as the number of packets in the previous minute and the time since the last packet. Data preprocessing is essential for dataset quality. Corrupted packets are removed, and missing values are handled using interpolation methods. To balance the dataset, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. All features undergo the Min-Max normalization technique to fit within a range of [0,1].

Table 1. Dataset Description

Dataset	Number of Entries (Approx.)	Number of Features	Classes and Scenarios
CICIDS2017	Flows: 2,830,540 Packets: 11,522,402	Flows: 80+	Multi-class scenarios include both benign and several types of malicious traffic.
NSL-KDD	~150,000	41	Normal, U2R, R2L, Probe, DoS

- **CICIDS2017**: The dataset contains flow-based and packet-based data, resulting in over 2.8 million flows and 11.5 million packets. Each flow is characterized by more than 80 features. Given its high dimensionality, multi-class labels, and class imbalances, CICIDS2017 is complex and challenging, making it particularly valuable for rigorous evaluations of IDS.
- **NSL-KDD**: An enhanced version of the KDD Cup '99 dataset, NSL-KDD resolves some of the older dataset's issues. With 41 features, it captures diverse network behaviors, classifying them into five primary categories: Normal (benign), U2R (User to Root attack), R2L (Remote to Local attack), Probe (surveillance and probing), and DoS (Denial of Service). Due to its assortment of attack types, NSL-KDD is frequently used for constructing and benchmarking IDS.

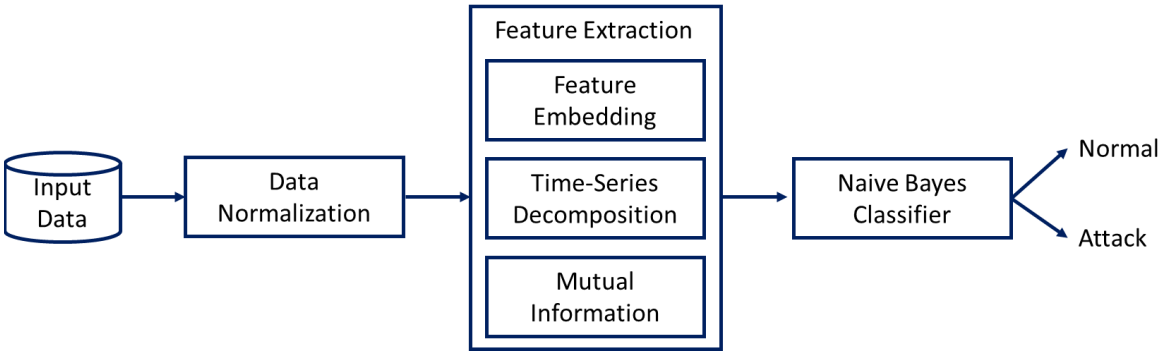

Fig 2. Proposed IDS Architecture

Proposed IDS Using Feature Extraction with the Naive Bayes Classifier (FE+NB) Model

Fig 2 depicts the proposed IDS model. Upon receiving input, the data undergoes preprocessing using Min-Max normalization, ensuring its readiness for the FE phase. This phase is critical, extracting high-quality features tailored for IoT traffic data. It starts with data compression via autoencoders, followed by time-series decomposition to identify inherent traffic patterns. A network topology analysis then provides a structural analysis of the IoT network, identifying potential vulnerabilities. The extracted features are further refined using the mutual information method, ensuring relevance for classification. With these features, the NB classifier is employed, calculating the probability of each instance being categorized as "normal" or "attack." The subsequent sections offer a detailed exploration of these FE techniques and the classifier, each with their respective algorithms.

Feature Extraction

Accurate threat detection in IoT networks hinges on the quality of features used in the classification models. This section outlines a sequential approach to FE tailored for IoT traffic data. The process starts with data compression, transitions through temporal and structural analyses, and concludes with FS based on their relevance to the classification task. Each stage addresses the unique challenges and characteristics of IoT network traffic.

Feature Embedding with Autoencoders

Initially, the dataset is subjected to feature embedding using autoencoders. At this stage, the high-dimensional raw input data is transformed into a more condensed representation. Autoencoders achieve this by attempting to replicate the original data. These neural networks reduce noise and redundancies, capturing only the most salient features. Autoencoders can transform raw input features into a more compressed representation. Let X be the input data matrix where each row represents an instance and each column a feature. The autoencoder's goal is to minimize the difference between X and its reconstruction $\hat{X}$ through encoding and decoding processes, EQU (3)

$$\hat{X} = f_{\text{decode}}\left(f_{\text{encode}}\left(X;\theta_e\right);\theta_d\right) \tag{3}$$

Where:

- f_{encode} represents the encoder function with parameters θ_e.
- f_{decode} represents the decoder function with parameters θ_d.

The loss, typically the Mean Squared Error (MSE), is given by EQU (4)

$$L(X,\hat{X}) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2 \tag{4}$$

Harnessing the power of autoencoders, the condensed data representation is now more apt for temporal analysis. The intent is to unearth patterns that unfold over time, which might be crucial indicators in IoT traffic.

Time-Series Decomposition for Temporal Features

The condensed data is subsequently decomposed to reveal inherent temporal structures. This decomposition breaks down the data into trend, seasonality, and residuals, offering a more granular view of traffic patterns. Identifying such patterns aids in discerning regular behavior from potential threats. Given a time series S_t, it is decomposed into three components: trend T_t, seasonality S_t, and residual R_t, EQU (5).

$$S_t = T_t + S_t + R_t \tag{5}$$

- **Trend Component:** Captured using moving averages.
- **Seasonality Component:** Extracted by analyzing periodic fluctuations in the data.
- **Residual Component:** It is the difference between the original series and the sum of the trend and seasonality components: $R_t = S_t - (T_t + S_t)$.

While temporal features present a sequential view of the data, understanding the broader structural relationships within the network offers insights into its vulnerability and dynamics. Having discerned temporal aspects, it is pivotal to assess the broader structural intricacies of the network. This step pinpoints vulnerabilities and reveals overarching dynamics.

Network Topology Analysis

The dataset, with its highlighted temporal features, is now mapped onto a network graph for structural analysis. Here, nodes represent devices or endpoints, while edges signify interactions. Exploring centrality measures and modularity can spotlight potential primary targets for network threats or identify tightly-knit communities in the network.

For a given network graph $G(V,E)$, where V is the set of vertices (nodes) and E is the set of edges:

Centrality Measures, EQU (6) to EQU (8)

$$\text{Degree Centrality: } C_D(v) = \frac{\text{degree}\,(v)}{|V|-1} \tag{6}$$

$$\text{Closeness Centrality: } C_C(v) = \frac{|V|-1}{\sum_{u\in V} d(v,u)} \tag{7}$$

$$\text{Betweenness Centrality: } C_B(v) = \sum_{s\neq v\neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{8}$$

Where $d(v,u)$ is the shortest path between nodes v and u, σ_{st} is the total number of shortest paths from node s to node t, and $\sigma_{st}(v)$ is the number of those paths that pass through v.

Cluster Analysis: The Modularity Q is used to measure the strength of the network's division into clusters or communities, EQU (9)

$$Q = \frac{1}{2m}\sum_{vw}\left[A_{vw} - \frac{k_v k_{ww}}{2m}\right]\delta(c_v, c_w) \tag{9}$$

Where A_{vw} is the adjacency matrix, k_v is the degree of node v, m is the number of edges, $\delta(c_v, c_w)$ is 1 if nodes v and w are in the same community and 0 otherwise.

After the structural exploration, the FEs are scrutinized to ascertain their relevance for the classification task. This ensures that only the most informative features steer the classification.

Mutual Information for Feature Selection

The Mutual Information metric evaluates the degree of dependency between each feature and the target classification. By gauging their shared information, features are ranked, ensuring that the most pertinent ones are prioritized in the detection process. Given two random variables X (features) and Y (classes), the mutual information $I(X;Y)$ is calculated as EQU (10)

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{10}$$

where $p(x,y)$ is the joint probability, and $p(x)$ and $p(y)$ are the marginal probabilities. Features with higher mutual information scores concerning the target class are more relevant and prioritised.

These methods extract a multi-faceted and comprehensive set of features, laying a robust foundation for the subsequent classification tasks.

Algorithm: FE based on Temporal-Structural Synthesis

Input:

- Dataset D with raw features

Output:

- Processed dataset D' with extracted features

Steps:

- Feature Embedding with Autoencoders
  - Initialize autoencoder with encoder function f_{encode} and decoder function $f_{\text{decode.}}$.
  - Train the autoencoder on dataset D to minimize the reconstruction loss.
  - Transform D using the trained encoder to obtain a compressed representation D_c.
- Time-Series Decomposition for Temporal Features
  - For each time series S_t in D_c :
  - Decompose S_t into trend T_t, seasonality S_t', and residual R_t.
  - Append T_t, S_t', and R_t to dataset D_c as new features.
- Network Topology Analysis
  - Represent the dataset D_c as a network graph $G(V,E)$.
  - For each node v in V :
- Compute Degree Centrality $C_D(v)$.
  - Compute Closeness Centrality $C_C(v)$.
  - Compute Betweenness Centrality $C_B(v)$.
- Perform cluster analysis to determine the modularity Q of $G(V,E)$.
  - Append centrality measures and Q to the dataset D_c as new features.
- Mutual Information for Feature Selection
  - For each feature X in D_c and target class Y :
  - Compute mutual information $I(X;Y)$.
  - Rank features based on their $I(X;Y)$ scores.
  - Select the top N features with the highest scores to form a dataset D'.
- Return the processed dataset D'.

NB Model Implementation for IDS Classification

The NB classifier is rooted in Bayes' theorem, which computes the probability of an event based on prior knowledge of related conditions. It gets its "naive" label from the assumption that every feature in the dataset is independent of the others when considering the class variable. Mathematically, Bayes' theorem can be framed as EQU (11)

$$P(Y = y \mid X = x) = \frac{P(X=x|Y=y)\times P(Y=y)}{P(X=x)} \tag{11}$$

Here, $P(Y = y \mid X = x)$ stands for the posterior probability, $P(X = x \mid Y = y)$ denotes the likelihood, $P(Y = y)$ is the class's prior probability, and $P(X = x)$ is the predictor's prior probability. In the realm of intrusion detection, Y symbolizes the class (e.g., "normal" or "attack"), and X represents the features derived from network traffic. To train the NB Classifier, the initial step is to calculate the prior probabilities for every class present in the training dataset using the EQU (12):

$$P(Y = y) = \frac{\text{Number of occurrences of class } y}{\text{Total number of instances}} \tag{12}$$

Once established, every feature X_i and for each class y, the likelihood is ascertained with EQU (13)

$$P(X_i = x_i \mid Y = y) = \frac{\text{Number of instances with } X_i{=}x_i \text{ in-class } y}{\text{Total number of instances in class } y} \tag{13}$$

Having set the foundational probabilities, the classification of a fresh instance with features x hinges on calculating the posterior probability for each possible class y. This is achieved by multiplying the prior probability of y with the likelihoods of all features given EQU (14)

$$P(Y = y \mid X = x) \propto P(Y = y) \times \prod_i P(X_i = x_i \mid Y = y) \tag{14}$$

The given instance is then attributed to the class y, which maximizes this posterior probability. After the classification process, the NB classifier outputs the probability scores for each class corresponding to the input instances. These scores indicate the likelihood of each instance belonging to a particular class. Consequently, by comparing these likelihoods, a definitive classification label, either "normal" or "attack", is assigned to each instance. This systematic probabilistic assessment provides a nuanced perspective, enabling the identification of potential threats and an understanding of their relative likelihood. This distinction can be instrumental in prioritizing responses in real-world intrusion detection scenarios.

Algorithm: Feature-based NB Classifier for IDS
Input: Training set D containing feature vectors $x^{(i)}$and their corresponding class labels $C^{(i)}$, Testing set S containing feature vectors.
Output: Classified labels for the testing set S.
Initialize Counters:
Create dictionaries (or arrays) for class priors $P(C^{(i)})$ and feature likelihoods $P(x_j \mid C^{(i)})$
Compute Class Priors:
For each class $C^{(i)}$ in D, EQU (15)

$$\text{Calculate } P(C^{(i)}) = \frac{\text{Number of occurrences of class } C^{(i)}}{\text{Total number of instances in } D} \tag{15}$$

Compute Likelihoods:
For each feature x_j in D and for each class $C^{(i)}$, EQU (16)

$$\text{Calculate } (x_j \mid C^{(i)}) = \frac{\text{Number of instances with } x_j \text{ in class } C^{(i)}}{\text{Total number of instances in class } C^{(i)}} \tag{16}$$

Classify Instances in Testing Set:
For each instance x in S :
Initialize max_posterior as negative infinity and predicted_class as null.
For Each class $C^{(i)}$, EQU (17)

$$\text{Compute posterior } P(C^{(i)} \mid x) \propto P(C^{(i)}) \times \prod_j P(x_j \mid C^{(i)}) \tag{17}$$

If posterior > max_posterior:
Max_posterior = posterior
Predicted_class = $C^{(i)}$
Assign predicted_class to instance x.
Return:
Return the classified labels for all instances in S.

IV. EXPERIMENTAL ANALYSIS

In the experimental analysis, the proposed model was tested on a system equipped with an Intel Core i7-10750H 6-Core Processor, 32 GB DDR4 RAM, 1TB NVMe SSD storage, and an NVIDIA GeForce RTX 2070 Super graphics card with 8GB

GDDR6. The software environment consisted of Ubuntu 20.04 LTS as the operating system, complemented by Python 3.8 for coding purposes. Key libraries like Scikit-learn were used for the NB implementation, while TensorFlow and Keras were employed for feature embedding tasks.

The dataset was partitioned into training and testing sets, with a 70:30 split ensuring a balanced representation. The training process for the NB model leveraged this training set. Key hyperparameters driving this training included a learning rate of 0.01, a batch size of 256 and 50 epochs, and a regularization parameter (alpha) of 0.001. These hyperparameters were selected based on a preliminary grid search to ensure optimized performance for the classifier.

The chosen evaluation metrics included Accuracy, which gauged the correct predictions made by the model; Precision, which indicated the ratio of True Positive (TP) predictions to the total positive predictions; Recall, which represented the number of true optimistic predictions against the total actual positives, F1-score, acting as the harmonic mean between precision and recall, and the ROC-AUC that demonstrated the model's ability to differentiate between the classes. To position the performance of this proposed NB classifier within a broader context, the results were compared against the following standard models, often utilized in intrusion detection:

Decision Trees: A non-parametric supervised learning method used for classification and regression. The decision tree maps features to outcomes, one decision at a time.

Random Forest: An ensemble learning method that creates a 'forest' from many decision trees. At the time of prediction, the mode of the decisions from individual trees is considered.

Support Vector Machines (SVM): A representation of examples as points in space, mapped so that separate categories are divided by a clear gap that is as wide as possible.

Standard NB: A probabilistic classifier based on Bayes' theorem with an assumption of independence between features. It is particularly suited for high-dimensional datasets.

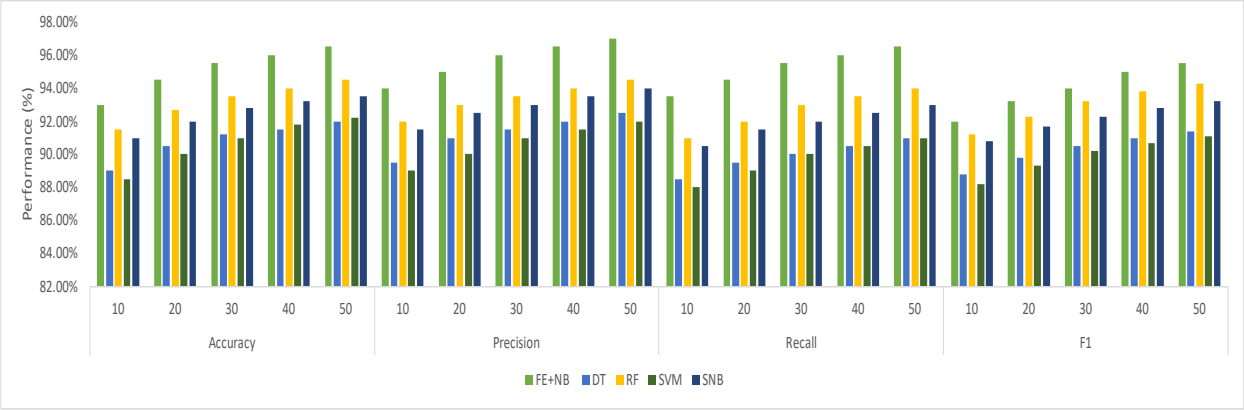

Fig 3. Precision across Epochs for Different Models

After training on the feature-engineered dataset, the NB classifier's performance was benchmarked against these models, considering the same features and experimental conditions. The outcomes serve as an indicator of where the proposed approach stands in comparison to established methods. **Fig 3** compares the model's performance against accuracy, precision, recall, and F1-score. When investigating the accuracy across epochs for the various models, it is evident that the FE+NB (proposed model) consistently outperforms its counterparts. Beginning with an accuracy of 93.0% at epoch 10, it exhibits a steady ascent to 96.5% by epoch 50. Comparatively, the Decision Trees and SVM demonstrate lower accuracies throughout, while the Random Forest and Standard NB models show competitive yet lower scores than the proposed model. By epoch 50, the gap in accuracy between the FE+NB and its closest competitor, the Random Forest, is a significant 2%. The precision metric presents a similar narrative. FE+NB starts at a robust 94.0% and elevates to an impressive 97.0% by epoch 50. Decision Trees and SVM, once again, lag behind the other models. RF's precision climbs reasonably but is outshined by the FE+NB model across all epochs. Notably, the Standard NB displays an admirable progression, though it does not surpass the proposed model's precision.

Regarding recall**,** FE+NB initiates with a 93.5% score at epoch 10 and culminates at 96.5% by epoch 50. All other models, including the Standard NB, start at a lower baseline and, despite improvements, fail to match or exceed the recall of the FE+NB model. This highlights the proposed model's capability to identify TP classifications accurately. When we analyze the F1-score**,** which balances precision and recall, the proposed model's prowess becomes even more evident. Beginning at a strong 92.0% at epoch 10, it peaks at 95.5% by epoch 50. The other models, though showing growth, do not challenge the supremacy of the FE+NB. Notably, the Decision Trees and SVM scores are consistently outpaced by both ensemble methods and the FE+NB model.

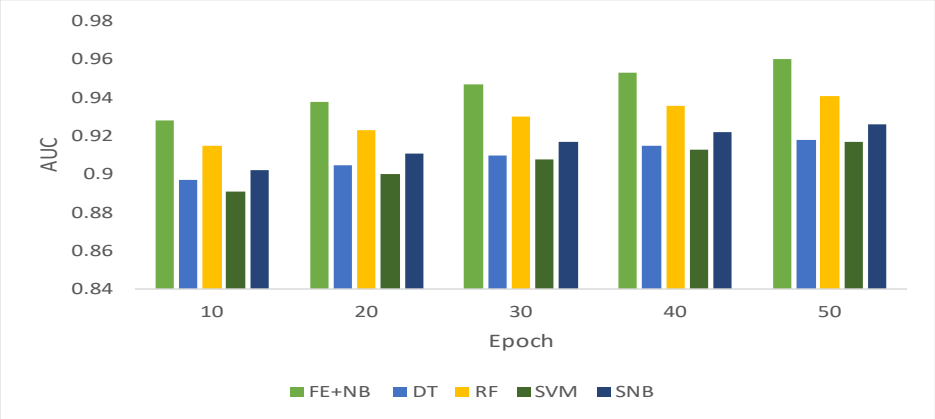

Fig 3. AUC across Epochs for Different Models

Lastly, as shown in **Fig 3**, the Area Under Curve (AUC) values, which represent the model's ability to differentiate between classes, reiterate the dominance of the proposed model. From an AUC of 0.928 at epoch 10 to 0.960 at epoch 50, the FE+NB model maintains a lead. While the Random Forest and Standard NB present commendable AUC values, neither surpasses the proposed model.

V. CONCLUSION AND FUTURE WORK

The realm of the Internet of Things (IoT) has experienced exponential growth, revolutionizing industries and everyday life. With this expansion, however, comes the imperative need for robust data security measures. While traditional Intrusion Detection Systems (IDS) offer a line of defense, their efficacy is often challenged by IoT data's sheer volume and complexity. Addressing this critical gap, the "FE+NB" model presents a pioneering approach, merging the strengths of sophisticated FE with the Naive Bayes (NB) classifier. Through rigorous experimentation and analysis, this model has demonstrated superior performance metrics, highlighting its potential to become a benchmark in IoT security.

As IoT continues its trajectory towards a more interconnected future, solutions like the "FE+NB" model stand as beacons, ensuring that data security and intrusion detection evolve in a cycle with technological advancements.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References

[1]. M. K. Thota, F. H. Shajin, and P. Rajesh, “Survey on software defect prediction techniques,” International Journal of Applied Science and Engineering, vol. 17, no. 4, pp. 331–344, 2020.

[2]. Y. V. R. N. Pawan, K. B. Prakash, S. Chowdhury, and Y.-C. Hu, “Particle swarm optimization performance improvement using deep learning techniques,” Multimedia Tools and Applications, vol. 81, no. 19, pp. 27949–27968, Mar. 2022, doi: 10.1007/s11042-022-12966-1.

[3]. A. Appathurai, R. Sundarasekar, C. Raja, E. J. Alex, C. A. Palagan, and A. Nithya, “An Efficient Optimal Neural Network-Based Moving Vehicle Detection in Traffic Video Surveillance System,” Circuits, Systems, and Signal Processing, vol. 39, no. 2, pp. 734–756, Aug. 2019, doi: 10.1007/s00034-019-01224-9.

[4]. N. Yuvaraj, T. Karthikeyan, and K. Praghash, “An Improved Task Allocation Scheme in Serverless Computing Using Gray Wolf Optimization (GWO) Based Reinforcement Learning (RIL) Approach,” Wireless Personal Communications, vol. 117, no. 3, pp. 2403–2421, Nov. 2020, doi: 10.1007/s11277-020-07981-0.

[5]. S. Deshmukh, K. Thirupathi Rao, and M. Shabaz, “Collaborative Learning Based Straggler Prevention in Large-Scale Distributed Computing Framework,” Security and Communication Networks, vol. 2021, pp. 1–9, May 2021, doi: 10.1155/2021/8340925.

[6]. E. Rajesh Kumar, K. V. S. N. Rama Rao, S. R. Nayak, and R. Chandra, “Suicidal ideation prediction in twitter data using machine learning techniques,” Journal of Interdisciplinary Mathematics, vol. 23, no. 1, pp. 117–125, Jan. 2020, doi: 10.1080/09720502.2020.1721674.

[7]. S. Stalin et al., "A Machine Learning-Based Big EEG Data Artifact Detection and Wavelet-Based Removal: An Empirical Approach," Mathematical Problems in Engineering, vol. 2021, pp. 1–11, Oct. 2021, doi: 10.1155/2021/2942808.
[8]. R. K et al., "A Robust And Accurate Video Watermarking System Based On SVD Hybridation For Performance Assessment," International Journal of Engineering Trends and Technology, vol. 68, no. 7, pp. 19–24, Jul. 2020, doi: 10.14445/22315381/ijett-v68i7p204s.
[9]. R. Janarthanan, R. U. Maheshwari, P. K. Shukla, P. K. Shukla, S. Mirjalili, and M. Kumar, "Intelligent Detection of the PV Faults Based on Artificial Neural Network and Type 2 Fuzzy Systems," Energies, vol. 14, no. 20, p. 6584, Oct. 2021, doi: 10.3390/en14206584.
[10]. S. Sengan, P. Vidya Sagar, R. Ramesh, O. I. Khalaf, and R. Dhanapal, "The optimization of reconfigured real-time datasets for improving classification performance of machine learning algorithms," Mathematics in Engineering, Science and Aerospace, vol. 12, no. 1, pp. 43–54, 2021.
[11]. D. Balamurugan, S. S. Aravinth, P. C. S. Reddy, A. Rupani, and A. Manikandan, "Multiview Objects Recognition Using Deep Learning-Based Wrap-CNN with Voting Scheme," Neural Processing Letters, vol. 54, no. 3, pp. 1495–1521, Apr. 2022, doi: 10.1007/s11063-021-10679-4.
[12]. S. Kumar, A. Jain, S. Rani, H. Alshazly, S. Ahmed Idris, and S. Bourouis, "Deep Neural Network Based Vehicle Detection and Classification of Aerial Images," Intelligent Automation & Soft Computing, vol. 34, no. 1, pp. 119–131, 2022, doi: 10.32604/iasc.2022.024812.
[13]. S. Rajasoundaran et al., "Secure routing with multi-watchdog construction using deep particle convolutional model for IoT based 5G wireless sensor networks," Computer Communications, vol. 187, pp. 71–82, Apr. 2022, doi: 10.1016/j.comcom.2022.02.004.
[14]. N. Yuvaraj, K. Praghash, R. A. Raja, and T. Karthikeyan, "An Investigation of Garbage Disposal Electric Vehicles (GDEVs) Integrated with Deep Neural Networking (DNN) and Intelligent Transportation System (ITS) in Smart City Management System (SCMS)," Wireless Personal Communications, vol. 123, no. 2, pp. 1733–1752, Oct. 2021, doi: 10.1007/s11277-021-09210-8.
[15]. S. C. Dharmadhikari, V. Gampala, Ch. M. Rao, S. Khasim, S. Jain, and R. Bhaskaran, "A smart grid incorporated with ML and IoT for a secure management system," Microprocessors and Microsystems, vol. 83, p. 103954, Jun. 2021, doi: 10.1016/j.micpro.2021.103954.
[16]. S. P. Jaiprakash, M. B. Desai, C. S. Prakash, V. H. Mistry, and K. L. Radadiya, "Low dimensional DCT and DWT feature based model for detection of image splicing and copy-move forgery," Multimedia Tools and Applications, vol. 79, no. 39–40, pp. 29977–30005, Aug. 2020, doi: 10.1007/s11042-020-09415-2.
[17]. J. R. K. K. Dabbakuti, A. Jacob, V. R. Veeravalli, and R. K. Kallakunta, "Implementation of IoT analytics ionospheric forecasting system based on machine learning and ThingSpeak," IET Radar, Sonar & Navigation, vol. 14, no. 2, pp. 341–347, Jan. 2020, doi: 10.1049/iet-rsn.2019.0394.
[18]. T. Chakravorti and P. Satyanarayana, "Non linear system identification using kernel based exponentially extended random vector functional link network," Applied Soft Computing, vol. 89, p. 106117, Apr. 2020, doi: 10.1016/j.asoc.2020.106117.
[19]. A. Paul and S. P. Maity, "Machine Learning for Spectrum Information and Routing in Multihop Green Cognitive Radio Networks," IEEE Transactions on Green Communications and Networking, vol. 6, no. 2, pp. 825–835, Jun. 2022, doi: 10.1109/tgcn.2021.3127308.
[20]. M. Sathya et al., "A Novel, Efficient, and Secure Anomaly Detection Technique Using DWU-ODBN for IoT-Enabled Multimedia Communication Systems," Wireless Communications and Mobile Computing, vol. 2021, pp. 1–12, Dec. 2021, doi: 10.1155/2021/4989410.
[21]. S. D. M. Achanta, T. Karthikeyan, and R. V. Kanna, "Wearable sensor based acoustic gait analysis using phase transition-based optimization algorithm on IoT," International Journal of Speech Technology, Sep. 2021, doi: 10.1007/s10772-021-09893-1.
[22]. Dr. B. Singh, P. Kavitha, R. Regin, Dr. K. Praghash, S. Sujatha, and Dr. S. S. Rajest, "Optimized Node Clustering based on Received Signal Strength with Particle Ordered-filter Routing Used in VANET," Webology, vol. 17, no. 2, pp. 262–277, Dec. 2020, doi: 10.14704/web/v17i2/web17029.
[23]. G. Rekha, S. Malik, A. K. Tyagi, and M. M. Nair, "Intrusion Detection in Cyber Security: Role of Machine Learning and Data Mining in Cyber Security," Advances in Science, Technology and Engineering Systems Journal, vol. 5, no. 3, pp. 72–81, 2020, doi: 10.25046/aj050310.
[24]. K. N. Reddy and P. Bojja, "A new hybrid optimization method combining moth–flame optimization and teaching–learning-based optimization algorithms for visual tracking," Soft Computing, vol. 24, no. 24, pp. 18321–18347, May 2020, doi: 10.1007/s00500-020-05032-1.
[25]. C. Banchhor and N. Srinivasu, "FCNB: Fuzzy Correlative Naive Bayes Classifier with MapReduce Framework for Big Data Classification," Journal of Intelligent Systems, vol. 29, no. 1, pp. 994–1006, Oct. 2018, doi: 10.1515/jisys-2018-0020.