

# An Ensemble Model for Smart User Consumption Analysis on Over-The-Top Applications

Madugu Jeremiah Omanga<sup>1</sup>, Hambali Moshood A.<sup>2</sup>, Nnamani Ugochukwu Bethel<sup>3</sup>, Kindness Bulus Gago<sup>4</sup>

*Department of Computer Science, Taraba State University Jalingo, Nigeria.<sup>1</sup>*

*Department of Computer Science Federal University Wukari, Nigeria.<sup>2</sup>*

*Department of Computer Science, Taraba State University Jalingo, Nigeria.<sup>3</sup>*

*Department of Computer Science, Taraba State University Jalingo, Nigeria.<sup>4</sup>*

[madugujeremiah4@gmail.com](mailto:madugujeremiah4@gmail.com)<sup>1</sup>, [hambali@fuwukari.edu.ng](mailto:hambali@fuwukari.edu.ng)<sup>2</sup>, [nnamaniugo14@gmail.com](mailto:nnamaniugo14@gmail.com)<sup>3</sup>,  
[kindnessgago@gmail.com](mailto:kindnessgago@gmail.com)<sup>4</sup>

## ABSTRACT

*Over-the-top (OTT) applications are known for large consumption of network resources to provide a range of services based on consumer's demand. The implementation of service degradation by network providers as a tool to address excessive data consumption patterns of OTT applications is consequently done universally rather than according to the user application's data consumption rate. Trends and patterns in consumption are seldom taken into consideration while executing degradation. This study aims to develop an ensemble model of Support Vector Machine (SVM) and Multilayer Perceptron (MLP) for smart user consumption analysis on OTT applications. Unicauca-dataset-April-June-2019-Network-flows, was used in this study to develop the proposed model. Due to the high dimensionality of the dataset, the Spearman rank correlation coefficient was used to extract the most relevant features from the dataset to enhance the model performance. Result from the implementation of the model revealed that, the ensemble model performed with a high overall accuracy of 99%, decent precision, recall, and F1-score. With the classification result from this study, the analysis of users' consumption trends and patterns may now be considered. Thereby enabling the implementation of service degradation on an application-by-application basis rather than in an all-inclusive manner.*

*Keywords: OTT application; network; service degradation; SVM; MLP*

## 1. Introduction

The widespread adoption of mobile communication devices is transforming the way people consume media globally (Filip et al., 2022). As technology progresses, these devices continually expand their functionalities to meet evolving consumer and stakeholder needs (James et al., 2023). Modern mobile devices now feature advanced multimedia capabilities, including larger screens, enhanced color displays, polyphonic sound, and video playback. These features have opened new avenues for mobile media use, particularly in advertising and various internet-driven activities that foster diverse user engagements (Ipek, 2020).

Digitalization has significantly altered media consumption patterns, particularly with the rise of Over-the-Top (OTT) media services, which deliver content directly via the internet. Enhanced networks, technological advancements, and the widespread availability of smart devices have fueled this trend. The COVID-19 pandemic further accelerated changes in media consumption behaviors and lifestyles (Gupta & Singharia, 2021). Consequently, streaming services now have unprecedented opportunities for real-

time interaction and communication with users, necessitating that service providers focus on the critical factors influencing customer retention and consistent revenue generation (Viswanathan et al., 2017).

OTT applications, such as Facebook, WhatsApp, Instagram, Telegram, TikTok, YouTube, Netflix, and Skype, utilize network operators' infrastructure to deliver audio, video, and other media services over the Internet without directly managing the content (Sudtasan & Mitomo, 2016). The increasing popularity of these apps is driven by the widespread use of smartphones and tablets and the affordability of internet packages (Jain, 2021). However, the extensive consumption of network resources by OTT apps to provide various services significantly impacts network management.

A common strategy employed by network providers to manage excessive data consumption is service degradation, which is often applied universally rather than tailored to specific applications, therefore lacking precise configuration. This approach rarely considers consumption trends and patterns (Chetty et al., 2015). To enhance network resource management and overall network performance, a comprehensive analysis of smart users' data consumption behavior is crucial (Omanga et al., 2023). Addressing this gap, this study proposes an ensemble model to predict smart users' network resource consumption on OTT applications, aiming to optimize network management strategies and improve service delivery.

## 2. Related Works

Shu (2016) presented a structure responsible for monitoring and analyzing continuous network traffic. The building's Internet Protocol (IP) and Software Define Network (SDN) were designed by the architect to include two SDN wired switches and a single regulator. Utilizing the Linked Layer Discovery Protocol (LLDP), which is run in an SDN, the suggested architecture was able to estimate the number of streams that are formed when network components interact and exchange data, all while changing the overall perspective of the network. Noughabi *et al.* (2016) developed a hybrid data mining strategy that used a priority-based approach to distribute network resources among users in a network by employing clustering and classification. This study aimed to distinguish itself from others by, predicting user behaviour in a network context, and investigate the primary drivers of internet browsing behaviour. Rojas *et al.* (2018) used the data packet inspection (DPI) method for data collection and processing. SOM (Self Organized Map) was used to do clustering analysis on the dataset to find user groups that have similar data consumption patterns. Random Forest, Boosting with J48, Bagging with J48, Support Vector Machines with LibSVM and SMO (Sequential Minimal Optimization) algorithms, J48, K-Nearest Neighbor (KNN), and Naive Bayes, were used as the machine learning model for classification. The outcome showed that the SMO trained with a linear kernel function is the best classifier. Goeffery *et al.* (2018) employed an unsupervised machine learning strategy and compared the results to those of a supervised machine learning approach that had previously been deployed. The supervised strategy employed the Naive Bayes classifier, while the unsupervised approach made use of an Expectation Maximization (EM) based clustering algorithm. The results demonstrated that the unsupervised clustering approach has an accuracy of up to 91% and performs up to 9% better than the supervised technique. Rolysent *et al.* (2018) proposed a Fuzzy-Based Dynamic Network Resource Allocation Approach that was developed for the Campus Area Network. The suggested method was implemented by combining the Mamdani fuzzy inference system with Web Usage Mining (WUM). Twenty concurrent users were used to test the prototype, and the results showed that it dynamically distributed network resources among users in accordance with the parameters. Rojas *et al.* (2019), compared incremental learning with traditional methods using a consumption behaviour analysis of OTT services. The test was conducted using two datasets: the first set had 131 characteristics and 1,581 occurrences. Every dataset instance is a user's consumption profile, which

contains condensed data on the user's consumption patterns about the 29 OTT apps found in the various IP flows.

The second dataset is comprised of 150,000 instances, each of which represents a synthetically constructed user's consumption profile. Support Vector Machine was the best algorithm from the conventional technique, according to an analysis of the results, while the ensemble method was the best classifier from the incremental approach. Choi *et al.* (2020) created a time-aware learning framework based on machine and deep learning skills for classifying OTT consumers. Utilizing an already-existing dataset of traffic that was collected from a university network and had 1581 occurrences and 130 attributes. According to the evaluation's findings, MLP and Convolutional Neural Network (CNN) fared better than the other models, with accuracy rates of 98.2 and 97.6%, respectively. Rojas *et al.* (2020) attempted to classify users' OTT consumption behavior based on Incremental Learning to personalize service degradation rules. They tested the efficacy of several incremental learning (IL) algorithms on a real world dataset of OTT applications to assess the performance of their model. Analyzing the outcome showed that Adaptive Random Forest (ARF) and the combination of Leverage Bagging (LB) and ARF performed the best, with 90% classification recall and precision. Jonathan *et al.* (2021) performed a categorization of network traffic using machine learning approaches from two different perspectives, one of which requires feature selection and the other does not.

Experimental results show that the featureless classification has an average accuracy and runtime of 94.14% and 0.52 seconds. The feature selection method, on the other hand, has an average duration of 0.25 seconds and an accuracy rate of 95.61%. Choi & Kim (2021) conducted an OTT consumer analysis that reflected the real market situation. The authors put forth a system that may categorize customers according to the real environmental circumstances of the OTT industry. The Synthetic Minority Over-sampling Technique (SMOTE) approach was employed to modify the dataset to better align with the real-world scenario. Models like J48-decision trees, KNN, and SVM were put into practice. They were tested on both balanced and refined datasets, and the outcomes were assessed. The models' results differed depending on the dataset that was utilized.

### 3. Research Methodology

To classify smart user consumption behaviour using an ensemble machine learning model, the following methodology was developed. This study utilized Unicauca-dataset-April-June-2019-Network-flows which is a comprehensive collection of network flow data captured at Universidad Del Cauca in Popayán, Colombia. The dataset consists of 2,704,839 flow occurrences, providing a detailed snapshot of network traffic over a three-month period. The data is saved in CSV format and includes a variety of features that describe the characteristics of each network flow. Here's a detailed breakdown of the dataset's features:

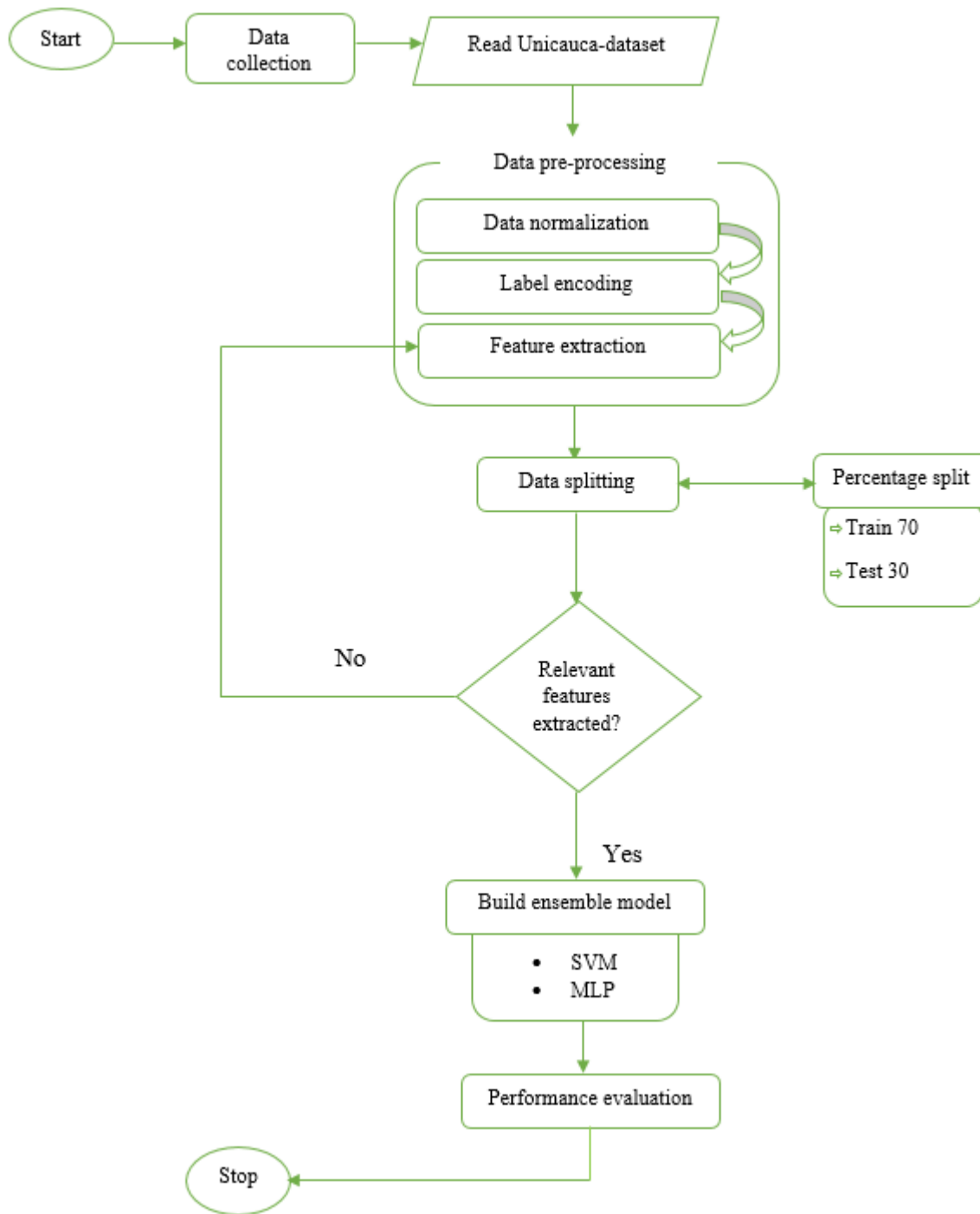
- i. **flow\_key**: This feature is a unique identifier for each flow, generated through a hash algorithm. It ensures that each flow can be distinctly identified.
- ii. **src\_ip\_numeric**: The source IP address in decimal format, which facilitates numerical operations and comparisons.
- iii. **src\_ip, dst\_ip**: These are the source and destination IP addresses in the standard network format, allowing for the identification of communication endpoints.
- iv. **src\_port, dst\_port**: These features represent the source and destination port numbers, which are essential for identifying specific services or applications involved in the communication.

- v. **Proto**: This denotes the transport protocol number as per IANA specifications (e.g., 1 for ICMP, 6 for TCP, 17 for UDP), indicating the protocol used for the network flow.
- vi. **pktTotalCount**: The total number of packets exchanged in both directions during the flow.
- vii. **octetTotalCount**: The total number of bytes exchanged in both directions, focusing solely on the IP payload.
- viii. **min\_ps, max\_ps**: These features represent the minimum and maximum packet sizes observed during the flow in both directions.
- ix. **avg\_ps**: The average packet size during the flow in both directions.
- x. **std\_dev\_ps**: The standard deviation of packet sizes during the flow in both directions, indicating the variability in packet sizes.
- xi. **flowStart, flowEnd, flowDuration**: These features record the start time, end time, and total duration of the flow in seconds, using UNIX time format.
- xii. **min\_piat, max\_piat, avg\_piat**: These features capture the minimum, maximum, and average packet inter-arrival times during the flow in both directions.
- xiii. **std\_dev\_piat**: The standard deviation of packet inter-arrival times during the flow in both directions, showing the variability in packet arrival intervals.
- xiv. **f\_pktTotalCount**: The total number of packets exchanged in the forward direction.
- xv. **f\_octetTotalCount**: The total number of bytes exchanged in the forward direction, focusing on the IP payload.
- xvi. **f\_min\_ps, f\_max\_ps, f\_avg\_ps, f\_std\_dev\_ps**: These features capture the minimum, maximum, average packet sizes, and standard deviation of packet sizes in the forward direction.
- xvii. **f\_flowStart, f\_flowEnd, f\_flowDuration**: These features record the start time, end time, and total duration of the flow in the forward direction in seconds using UNIX time format.
- xviii. **f\_min\_piat, f\_max\_piat, f\_avg\_piat, f\_std\_dev\_piat**: These features capture the minimum, maximum, and average packet inter-arrival times and their standard deviation in the forward direction.
- xix. **b\_pktTotalCount**: The total number of packets exchanged in the backward direction.
- xx. **b\_min\_ps, b\_max\_ps, b\_avg\_ps, b\_std\_dev\_ps**: These features capture the minimum, maximum, and average packet sizes, and standard deviation of packet sizes in the backward direction.
- xxi. **b\_flowStart, b\_flowEnd, b\_flowDuration**: These features record the start time, end time, and total duration of the flow in the backward direction in seconds using UNIX time format.
- xxii. **b\_min\_piat, b\_max\_piat, b\_avg\_piat, b\_std\_dev\_piat**: These features capture the minimum, maximum, and average packet inter-arrival times and their standard deviation in the backward direction.

- xxiii. **flowEndReason**: This feature indicates why the flow was terminated and sent to the final array, with various reasons such as inactive timeout, active timeout, forced expiration, FIN or RST flag detection, or timer expiration.
- xxiv. **text\_formatcategory**: The category of the communication as identified by nDPI (Network Deep Packet Inspection).
- xxv. **text\_formatapplication\_protocol**: The application protocol for the flow (e.g., TLS, HTTP, DNS) as detected by nDPI.
- xxvi. **text\_formatweb\_service**: The web service detected by nDPI (e.g., Facebook, WhatsApp, Google).

When dealing with network flow data, particularly data that includes IP addresses and port numbers, ethical considerations and user privacy protections are paramount. Data minimization and transparency were taken.

Reading and filtering the aforementioned dataset constitutes the initial stage. Data filtration was used to handle missing values and eliminate noisy data. Label encoding was then carried out after which the dataset was converted to a range of values between 0 and 1. Considering the high dimensionality of the dataset the Spearman rank correlation coefficient was employed to extract pertinent features from the dataset. In other to feed the ensemble machine learning model, the dataset was divided into the training set to train the model and the test set to validate the model. Fig. 1 presents the methodology framework for the research.



**Fig. 1:** Methodology Framework

### 3.1 Support Vector Machine



The support vector machine (SVM) is a widely used and effective technique in the big data for handling classification challenges. It uses a small number of training observations known as support vectors to identify a linear decision boundary that maximizes the distance between two classes in a feature space converted by a kernel function. The SVM technique uses hyperplanes to classify a given set of objects and is based on the concept of decision planes. SVM can classify new data after receiving a collection of labeled data. Drawing a border or decision boundary that splits n-dimensional space into classes is the aim of the SVM technique, which enables subsequent data points to be promptly allocated to the correct category. The ideal decision boundary is a hyperplane. The hyperplane is created by selecting the extreme vectors and points.

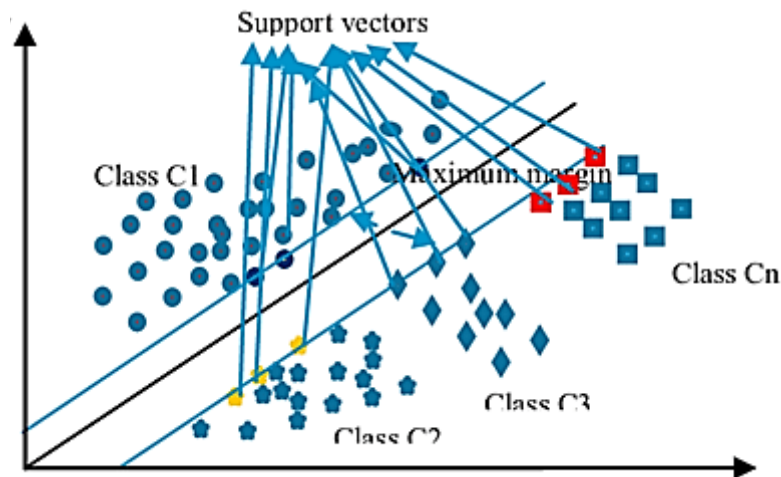


Fig. 2: SVM multiclass classification using decision hyperplane (Source: Mourya *et al.*, 2020)

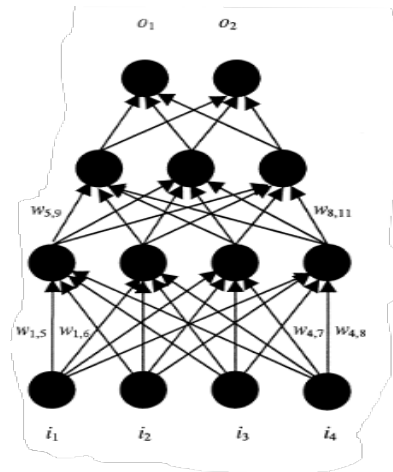
### 3.2 Multilayer Perceptron

With the help of a condensed collection of biological neural system principles, the multilayer perceptron deep learning method simulates a biological neural network (Worden *et al.*, 2023). Using vectors of organized perceptions, MLP uses the output of one layer as the input for subsequent levels and perhaps even higher layers. Because of this, a perceptron may be linked to all or some of the senses in the layer below, and these links mimic the synaptic connections in the brain. By means of a perceptron, a weighted data signal simulates the electrical stimulation of a nerve cell and, as a result, the information transmission of the network or brain. A connection weight,  $w_m$ ,  $m$ , multiplies the input values to a processing element to imitate the network's neural pathways becoming stronger (Ivanov *et al.*, 2022). As such, the model's learning process is simulated by varying the connection strengths or weights, and a scalar function is used to aggregate all of the weight-adjusted input values from a processing unit such that :

$$y = \sum w_{ij}x_i \tag{1}$$

Equation 1's summation gives the perceptron a single input value. Thus, after computing an input value, the processing element generates its output (and subsequently the input signals for the subsequent processing layer) using a transfer function. The perceptron's input value is transformed by the transfer function. This transformation usually uses the activation function, which is a non-linear function. Up until the neural network produces a final output value, the procedure is repeated between layers of processing

units until the value  $O_n$  is formed by the neural network. Fig. 2 illustrates the processing mechanism of MLP.



**Fig. 2: Multilayer Perceptron** (source: Grosse, 2019)

### 3.3 Ensemble Model

Using the stacking strategy, an ensemble of a support vector machine (SVM) and multilayer perceptron (MLP) was developed in this study with SVM as the base learner. The developed model was validated using the test data. This was to ascertain the performance of the model in the classification of OTT application user consumption behaviour. A confusion matrix was used to evaluate the performance of the smart user consumption prediction ensemble model.

---

**Algorithm 3.5:** Ensemble model’s pseudocode

---

*Stacking:*

*Input:*

*Data set  $D = (x_1, y_1), (x_2, y_2), \dots, (x_M, y_M)$ ;*

*Base learning algorithm  $L_t$  ( $t = 1, 2, 3, \dots, T$ );*

*Meta learning algorithm  $L$ ;*

*Process*

---



---

```

For t = 1,2,...,T:
    Ht = L(Dt); # Train base learners ht by applying the level-0
End
D' = ∅; # Creat a new dataset
For m = 1,2,...,M;
    For t = 1,2,...,T:
        zit = ht(xi); # use ht to classify the training example xi
    End;
    D' = D' U {((z1i, z2i, ..., zti), yi)}; # A new data set is finish
End;
h' = L(D'); # Train meta-learner h' by applying the level -1
Output:
H(x) = h'(h1(x), h2(x), ..., hr(x))
    
```

---

This ensemble model consists of base learning algorithm SVM denoted as  $L_t$ , a meta-learning algorithm MLP represented as  $L$ , and a Unicauca-dataset-April-June-2019-Network-flows denoted by  $D$  containing input-output pairs  $(x, y)$ .

The algorithm begins by iterating through the base learning algorithms  $L_t$  from  $t = 1$  to  $T$ . For each base learner, denoted as  $h_t$ , it applies the meta-learning algorithm  $L$  to train the base learner on the dataset  $D_t$ , which is a subset of the original dataset  $D$ . This process helps create a set of diverse base learners  $H_t$ , each trained on a different portion of the data or with different configurations.

After training all the base learners, the algorithm proceeds to create a new dataset, denoted as  $D'$ . It does this by iterating through each example  $(x, y)$  in the original dataset  $D$ . For each example, it then further iterates through the base learners ( $t = 1$  to  $T$ ) to obtain predictions  $z_{it}$  using the respective base learner  $h_t$ . These predictions  $z_{it}$ , along with the true output  $y_t$ , are used to create a new entry in the dataset  $D'$ .

Once the creation of  $D'$  is complete, the algorithm proceeds to train a meta-learner, denoted as  $h'$ , using this augmented dataset  $D'$ . The meta-learner is designed to learn from the predictions made by the base learners on the original dataset, effectively learning how to combine their outputs to make a final prediction. Finally, the output of this ensemble model, denoted as  $H(x)$ , is defined as the prediction made by the trained meta-learner  $h'$ . It takes as input the predictions of all the base learners  $h_1(x), h_2(x), \dots, h_r(x)$ , where  $r$  represents the total number of base learner.

### 3.4 Metrics of Evaluation

To evaluate the performance of the ensemble model, this study used accuracy, precision, recall, and F1-score as evaluation metrics as illustrated in equations 2, 3, 4, and 5. These metrics are calculated as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN \text{ (total sample)}} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

$$F1\_Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Where P denotes total positive prediction, and N denotes total negative prediction. Accuracy determines the classifier's recognition rate whereas, the degree of accuracy with which the classifier predicts is known as precision. The classifier's sensitivity in prediction is determined by recall. Finally, the F-score yields a single score by combining the recall and precision values and defining the harmonic mean of the model's accuracy and recall.

#### 4. Result and Discussion

The experimental setup for this study consists of a 64-bit Windows operating system with an Intel(R) Corel (TM) i5-3630QM CPU running at 2.40 GHz with 4.00 GB. The Application Programming Interface (API) called Sklearn was used in conjunction with a few other Python dependencies, including the Matplotlib library for data visualization tasks, NumPy for vector operations, pandas for file reading, and other data manipulations. Ensuring that the features in the dataset are scaled to a value between 0 and 1 at the data pre-processing stage, the standard scalar functionality was employed. Efficiently scaling the dataset characteristics is a great way to accelerate the convergence of the model. Spearman rank correlation coefficient feature selection technique was applied to select the most relevant features in the dataset due to the high dimensionality of the dataset utilized. Selected features are illustrated in Fig. 3.

```
Out[153]: Index(['src_ip', 'dst_ip', 'proto', 'pktTotalCount', 'octetTotalCount',
               'min_ps', 'max_ps', 'avg_ps', 'std_dev_ps', 'flowStart', 'flowEnd',
               'min_piat', 'max_piat', 'avg_piat', 'std_dev_piat', 'f_pktTotalCount',
               'f_octetTotalCount', 'f_min_ps', 'f_max_ps', 'f_avg_ps', 'f_std_dev_ps',
               'f_flowStart', 'f_flowEnd', 'f_flowDuration', 'f_min_piat',
               'f_max_piat', 'f_avg_piat', 'f_std_dev_piat', 'b_pktTotalCount',
               'b_octetTotalCount', 'b_min_ps', 'b_max_ps', 'b_avg_ps', 'b_std_dev_ps',
               'b_flowStart', 'b_flowEnd', 'b_min_piat', 'b_max_piat', 'b_avg_piat',
               'b_std_dev_piat', 'flowEndReason', 'category', 'application_protocol',
               'web_service'],
              dtype='object')
```

**Fig. 3:** List of the selected features via the Spearman rank correlation feature selection techniques.

The ensemble model utilized a learning rate of 0.0001, with Adam set as the optimizer and drop out of 0.2 for each of the layers for MLP. The optimizer, Adam, is used and is in charge of modifying the weights and biases in the model to increase accuracy and performance. Rectified Linear Unit (RELU) was applied to the input and hidden layer while SoftMax was used to the output layer due to the large number of class labels. The Kernel Cache Size has a strong impact on run times for the applied SVM algorithm. It determined the cache size needed for the program in the RAM (Random Access Memory), the cache size used for SVM was set to 200. Considering that the variables from the dataset are linear, the linear kernel was used. The result obtained from classifying smart users' consumption profiling using the ensemble model is presented in Fig. 4.

```

===== Ensemble Model =====
              precision    recall  f1-score   support

     0         0.99         1.00         0.99         7299
     1         0.98         0.81         0.89         506

 accuracy                   0.99         7805
 macro avg                 0.98         0.91         0.94         7805
 weighted avg              0.99         0.99         0.99         7805
    
```

**Fig. 4:** Ensemble model evaluation matrix result

The result revealed the ensemble model performed with a classification accuracy of 0.99%, precision of 0.99%, recall of 1.0% and f1\_score of 0.99% when trained and tested with the OTT application network traffic dataset. By implication, the ensemble model performed well overall in differentiating between various forms of network traffic, as seen by the reported accuracy of 0.99%, which indicates that 99% of the occurrences in the dataset were correctly classified. The model exhibited a high percentage of accurately classified positive occurrences (i.e., true positives) relative to the total instances predicted as positive, with a precision of 0.99%. The model's capacity to prevent false positives is reflected in this measure, which is important in situations where misclassification might have serious repercussions. The recall score of 1.0 indicates that the model successfully identified all instances of positive class (true positives) in the dataset. In other words, there were no instances of the positive class that were incorrectly classified as negative (false negatives). This metric is important in scenarios where capturing all positive instances is critical. The F1 score of 0.99% represents the harmonic mean of precision and recall, providing a balanced measure of the model's performance. The high F1 score indicates both high precision and recall, reflecting the model's ability to achieve accurate and comprehensive classification results.

**5. Conclusion and Recommendation**

This study developed an ensemble model combining a machine learning algorithm (SVM) and a deep learning algorithm (MLP) to classify smart users' consumption profiles using an OTT dataset sourced from Kaggle. The Unicauca-dataset-April-June-2019-Network-flows, containing 2,704,839 flow occurrences gathered from Universidad Del Cauca in Popayán, Colombia, was saved as a CSV. The dataset's size, variable types, and dimensionality influenced the chosen approach. The ensemble model demonstrated a high overall accuracy of 99%, with good precision, recall, and F1-score for both classes, as illustrated in Fig. 4. However, there is room for improvement, particularly in accurately recognizing occurrences of class 1, as the recall for class 1 is somewhat lower than for class 0.

Future research could explore various avenues to enhance the model's performance further. One potential direction is the integration of additional machine learning and deep learning techniques to improve the classification of class 1 occurrences. Ensemble methods could be expanded to include more diverse algorithms, potentially boosting overall accuracy and recall. Additionally, refining feature engineering processes and employing advanced dimensionality reduction techniques might uncover more significant patterns within the dataset, thereby enhancing model performance. By addressing these future directions, the ensemble model for smart user consumption analysis on OTT applications can be further refined, leading to more accurate and reliable classifications. This advancement would support more precise

degradation management on an application-by-application basis, enhancing the overall quality of service for users.

## 6. Contribution to knowledge

This research contributes to the field by demonstrating the efficacy of ensemble models in smart user consumption analysis on OTT applications. It highlights the potential of such models to achieve high performance metrics, thus providing a robust tool for network traffic classification. This contribution is significant as it provides a reliable method for analyzing and interpreting user consumption patterns in OTT applications, which can have applications in network management, user experience enhancement, and security monitoring. By validating the model's performance with impressive statistical metrics, the research sets a benchmark for future studies and applications in this area.

## Reference

- Chetty M, Kim H, Sundaresan S, Burnett S, Feamster N, Edwards WK., (2015). ucap: An internet data management tool for the home. In Proceedings of the 33rd annual ACM conference on human factors in computing systems. 3093-3102.
- Choi, J., and Kim, Y. (2021). A heterogeneous learning framework for over-the-top consumer analysis reflecting the actual market environment. *Applied Sciences*, *11*(11), 4783.
- Filip, R., Gheorghita Puscaselu, R., Anchidin-Norocel, L., Dimian, M., & Savage, W. K. (2022). Global challenges to public health care systems during the COVID-19 pandemic: a review of pandemic measures and problems. *Journal of personalized medicine*, *12*(8), 1295.
- Goeffery E, Adrian M, Aizoman A. (2014). Development of unsupervised machine learning Model for Internet Traffic Identification, Conference paper submitted to Department of Computer Science University of Calgary Calgary, AB, Canada T2N 1N4.;22-30. DOI: 10.1109/GLOCOM.2006.443
- Grosse, R. (2019). Lecture 5: Multilayer Perceptrons. *inf. t c.*
- Gupta, G., & Singharia, K. (2021). Consumption of OTT Media Streaming in COVID-19 Lockdown: Insights from PLS Analysis. *Vision*, *25* (1), 36–46.
- Ipek, I. (2020). Understanding consumer behavior in technology-mediated spaces. *Digital transformation in business and society: Theory and cases*, 169-189.
- Ivanov, D., Chezhegov, A., & Larionov, D. (2022). Neuromorphic artificial intelligence systems. *Frontiers in Neuroscience*, *16*, 959626.
- Jain MK., (2021).The Rise of OTT Platform: Changing Consumer Preferences. *EPR International Journal of Multidisciplinary Research (IJMR)*; (7)6:257-261.
- James, O., Olajide, O. B., & Omanga, M. J. (2023). Over-the-top Applications Traffic Analysis Model for Networks using Multilayer Perceptron (MLP) and Long Short-term Memory (LSTM). *Asian Journal of Pure and Applied Mathematics*, 242-250.
- Mourya, A. K., Kaur, H., & Uddin, M. (2020). A novel approach to heterogeneous multi-class SVM classification. In *New Paradigm in Decision Science and Management: Proceedings of ICDSM 2018* (pp. 39-47). Springer Singapo
- Noughabi, E. A., Far, B. H., and Raahemi, B. (2016). Predicting students' behavioral patterns in university networks for efficient bandwidth allocation: A hybrid data mining method (application paper). 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), Pittsburgh, PA, USA. <https://doi.org/10.1109/iri.2016.21>

- Omanga, M. J., Olajide, O. B., James, O., & Wreford, A. I. (2023). Smart User Consumption Analysis Using Support Vector Machine and Multilayer Perceptron. *Asian Research Journal of Current Science*, 163-170.
- Rojas JS, Gallón ÁR, Corrales JC., (2018). Personalized service degradation policies on OTT applications based on the consumption behavior of users. In International Conference on Computational Science and Its Applications. 543- 557.
- Rojas, J. S., Pekar, A., Rendón, Á., and Corrales, J. C. (2020). Smart user consumption profiling: Incremental learning-based OTT service degradation. *IEEE access*, 8, 207426-207442.
- Rojas, J. S., Rendon, A., and Corrales, J. C. (2019). Consumption behavior analysis of over-the-top services: Incremental learning or traditional methods? *IEEE Access*, 7, 136581-136591.
- Shu Z. (2016). "Traffic engineering in software-defined networking: Measurement and management," *IEEE Access*, vol. 4, pp. 3246–3256.
- Sudtasan T, Mitomo H., (2016). Effects of OTT services on consumer's willingness to pay for optical fiber broadband connection in Thailand. presented at the 27th Eur. Regional Conf. Int. Telecommun. Soc. (ITS), Evol. North-South Telecommun. Divide, Role Europe, Cambridge, U.K.; [Online]. Available: <https://www.econstor.eu/handle/10419/148709>
- Viswanathan, V., Hollebeek, L. D., Malthouse, E. C., Maslowska, E., Jung Kim, S., & Xie, W. (2017). The dynamics of consumer engagement with mobile technologies. *Service Science*, 9(1), 36–49.
- Worden, K., Tsialiamanis, G., Cross, E. J., & Rogers, T. J. (2023). Artificial neural networks. In *Machine Learning in Modeling and Simulation: Methods and Applications* (pp. 85-119). Cham: Springer International Publishing