IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Detection Of Anomalies In Text Messaging Using Machine Learning Techniques

[1]Mr. Raghuram A S, [2]Nithin M Singh, [3]Prajwal C, [4]Pramod Shanmukh Walluwine, [5]Santosh M

[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student

Dept. of Computer Science and Engineering,

ATME College of Engineering, Mysuru, India

Abstract: Everyday millions of people and hundreds of thousands of institutions communicate with each other over the Internet. In the once two decades, the number of people using the Internet has increased veritably presto. resemblant to these developments, the number of attacks made on the Internet is adding day by day. Although hand- grounded styles are used to help these attacks, they're unproductive against zero-day attacks. On the other hand, the Anomaly- grounded approach is an indispensable result to the network attacks and has the capability to descry zero- day attacks as well. In this study, it's aimed at detecting network anomaly using machine literacy styles. In this environment, the sample data has been used as a dataset because of its over- to currentness, and wide attack diversity. On this dataset, point selection was made by using the Random Forest Regressor algorithm. Seven different machine learning algorithms have been used in the operation step and achieved high performance.

Index Terms – Email Detection, Spam Detection, NLP, Machine Learning, Model Training, Message Testing.

I. INTRODUCTION

Dispatch has come an essential communication tool, extensively used for both particular and professional purposes. still, the swell in spam emails — unasked and frequently dangerous dispatches — poses significant challenges for druggies and associations likewise. Spam emails can range from promotional announcements and phishing attempts to dispatches containing vicious links or attachments, causing productivity loss, security pitfalls, and fiscal damage. Machine literacy- grounded dispatch spam discovery systems influence algorithms to dissect patterns in dispatch data and classify dispatches as" spam" or" ham"(non-spam). These systems can learn from large datasets and acclimatize to new types of spam by continuously streamlining their models. By rooting features from dispatch content, heads, and metadata, ML models can identify subtle pointers of spam that are frequently missed by traditional approaches. The development of a machine literacy- powered spam discovery system involves several stages, including data preprocessing, point birth, model training, and evaluation. By employing advanced ways similar as natural language processing (NLP), supervised literacy, and ensemble styles, these systems can achieve high delicacy while minimizing false cons and negatives. This design aims to produce an effective, scalable, and adaptive dispatch spam discovery system using machine literacy ways, addressing the evolving challenges posed by spam emails and icing a safer dispatch experience for druggies

II. OBJECTIVE

The ideal of this system is to develop a web- grounded operation that directly classifies textbook dispatches as" Spam" or" Not Spam" using machine literacy. The platform will give an intuitive stoner interface for data preprocessing, model training, and bracket testing. It'll integrate advanced textbook analysis ways, including emoji analysis, to enhance spam discovery delicacy. By using machine literacy algorithms, the system aims to ameliorate filtering effectiveness, minimize false cons, and acclimatize to evolving spam patterns. The thing is to offer a dependable and stoner-friendly tool for individualities and businesses to guard their dispatches against unwanted and potentially dangerous dispatches.

III. PROPOSED SYSTEM

The proposed email spam detection system is built using Flask as the web framework, with deployment options on local servers or cloud platforms like AWS, GCP, or Heroku. It supports MySQL as an optional database for extended functionality and uses pip within a virtual environment for dependency management. To enhance security and usability, the system includes authentication for controlled access to training and testing routes. Users can upload custom datasets to train the model, improving adaptability to specific needs. Advanced emoji analysis is incorporated using specialized libraries or custom embeddings to detect spam more effectively. Performance evaluation features such as a confusion matrix and visualizations help assess and refine the system's accuracy. Compared to existing solutions, this system significantly enhances productivity by filtering out spam emails, allowing users to focus on important messages without distractions. By blocking phishing attempts, malware, and other malicious content commonly found in spam emails, it strengthens cybersecurity and minimizes risks such as data breaches, financial losses, and reputational harm. The system also ensures compliance with privacy and data protection regulations by reducing exposure to harmful or sensitive content, helping organizations maintain regulatory standards and safeguard user data.

One of the key advantages of this system is its use of machine learning algorithms, which enable continuous improvement in spam detection accuracy. Unlike traditional rule-based systems that require constant manual updates, machine learning models adapt to evolving spam techniques, ensuring long-term effectiveness. As spam methods become more sophisticated, the system refines its detection capabilities, making it a reliable solution for email security. Additionally, by leveraging automation, the system reduces the manual effort required to filter spam, optimizing resource efficiency for individuals and organizations.

Overall, this spam detection system combines automation, security, adaptability, and compliance to provide a comprehensive solution for managing unwanted emails. Its integration with cloud platforms ensures scalability, while authentication features enhance access control. With the ability to train on custom datasets and utilize advanced emoji analysis, the system goes beyond traditional spam filtering methods. The inclusion of performance evaluation metrics helps users understand and improve detection accuracy over time. By offering a proactive defense against threats and continuously learning from new data, this system provides a modern and effective approach to email security.

IV. METHODOLOGY

The spam detection system is implemented using Flask and machine learning, leveraging natural language processing (NLP) and emoji-based analysis. It features a user-friendly web interface where users can train the model with new data or test custom messages. The system follows a structured methodology, starting with library setup, where essential dependencies like Flask, Scikit-learn, and NLTK are initialized. The preprocessing pipeline cleans text by converting to lowercase, removing punctuation, extracting emojis, and applying lemmatization to ensure high-quality input for training. Feature engineering is performed using TF-IDF vectorization for text and emoji frequency analysis, combining them into a numerical feature matrix. The model is trained using a Random Forest Classifier, with data split into training and testing sets for evaluation. Performance metrics like accuracy and classification reports help assess model efficiency. The Flask web application provides routes for training and testing, allowing real-time spam detection. Model persistence using pickle ensures that the trained model and vectorizer can be reused without retraining. Potential enhancements include integrating a database, using real-world datasets, experimenting with advanced models, improving error handling, and refining the user interface for a better experience.

Additional Enhancements:

- Database Integration: Store user messages and predictions using MySQL.
- Real-World Data: Train the model with actual spam datasets for improved accuracy.
- Advanced Models: Explore deep learning techniques to enhance spam detection.
- Error Handling: Implement robust mechanisms to handle user input and file errors.
- User Interface: Upgrade HTML templates for a more intuitive user experience.

V. IMPLEMENTATION

The email spam detection system is implemented using Flask and machine learning, incorporating text preprocessing, emoji analysis, and model training. The system collects labeled spam and non-spam emails, preprocesses them by removing punctuation, stop words, and extracting emojis, then converts text into numerical features using TF-IDF vectorization. A Random Forest classifier is trained on these features, and its performance is evaluated using accuracy, precision, recall, and F1-score. The model is saved using pickle for later use. The Flask web application provides three main routes: **Home**, which serves as the main interface; **Train Model**, allowing users to train the system with new data and view performance metrics; and **Test Message**, where users input an email to check if it is spam. The model predicts based on text and emoji features, displaying results in real time. The system continuously improves by retraining new data, ensuring adaptability to evolving spam patterns. Key enhancements include database integration for storing user messages, advanced deep learning models for improved accuracy, enhanced error handling, and a more user-friendly interface. This tool provides a practical and efficient way to classify spam messages while improving email security.

Flowchart: The email spam detection process starts when an email is received in the inbox. The system preprocesses the email by removing unnecessary formatting, converting it to plain text, and breaking it into words. Key features such as word frequency, spam-like keywords, sender details, and suspicious links are extracted. A trained classification model (e.g., Naive Bayes or SVM) analyzes these features to predict whether the email is spam. If the spam probability exceeds a set threshold, the email is marked as spam and moved to the Spam folder; otherwise, it remains in the inbox. The process then ends.

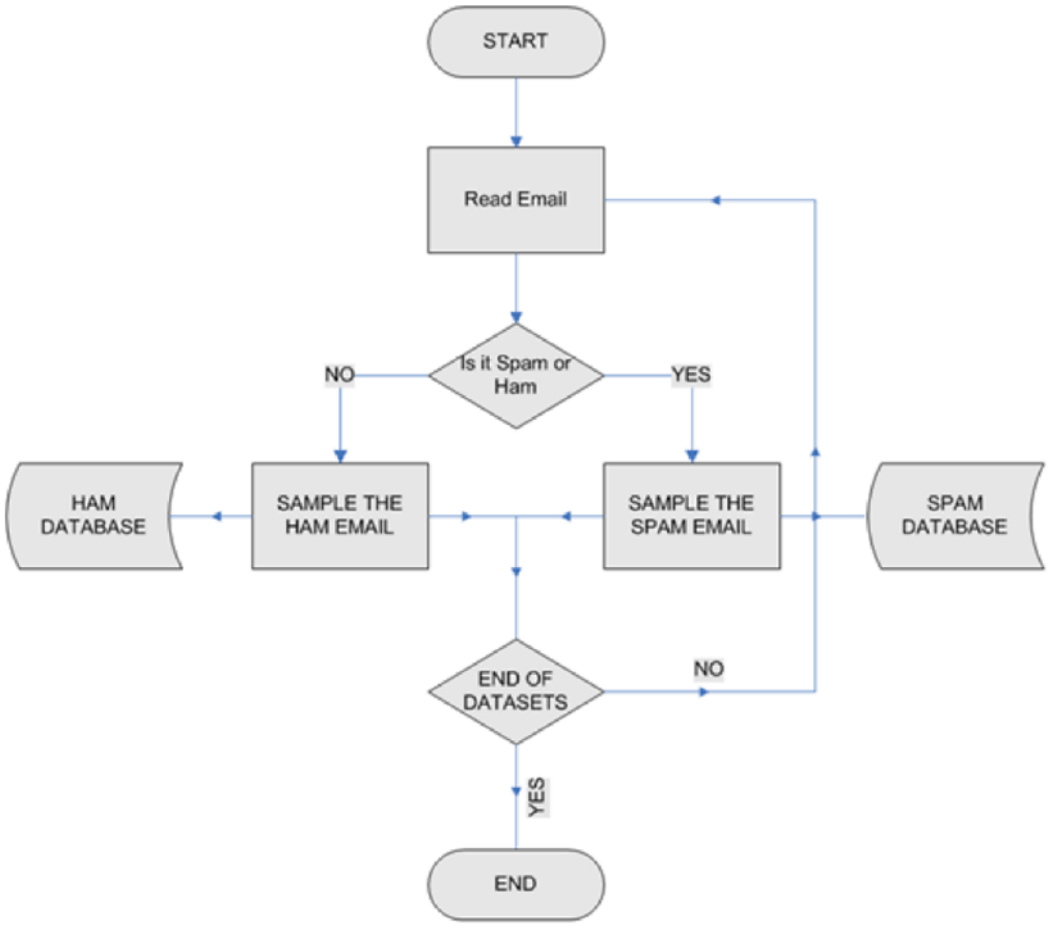

Figure 1: Working of Email Detection System

VI. RESULTS

The **Email Spam Detector** follows a structured process to ensure accurate spam classification. Users begin at the **Home Page**, which provides an overview of the system's features and performance metrics. The next step is the **Train Model** section, where users upload labeled datasets to train the system. The model learns patterns from spam and ham (genuine) messages and generates key performance metrics such as accuracy, precision, recall, and F1-score to assess its effectiveness.

Once the training is complete, users navigate to the **Test Message** section, where they input an email message for classification. The trained model analyzes the text and determines whether the message is **spam** (fake) or **ham** (genuine) based on the patterns learned. The system provides results in real time, allowing users to efficiently manage emails and reduce unwanted spam.

The model's performance is further evaluated using **macro average** and **weighted average scores**, ensuring balanced classification. If needed, users can refine and retrain the model for improved accuracy. By leveraging advanced machine learning techniques, the Email Spam Detector ensures **efficient, user-friendly, and reliable** spam detection.

Key Takeaways:

- **Home Page:** Central interface with system features and performance overview.
- **Train Model:** Upload datasets, train the model, and view performance metrics.
- **Test Message:** Enter a message to classify as spam or ham.
- **Performance Metrics:** Accuracy, precision, recall, and F1-score for evaluation.
- **Refinement & Retraining:** Improve model accuracy with new data.

Figure 2: This snapshot displays the home page of the email spam detector

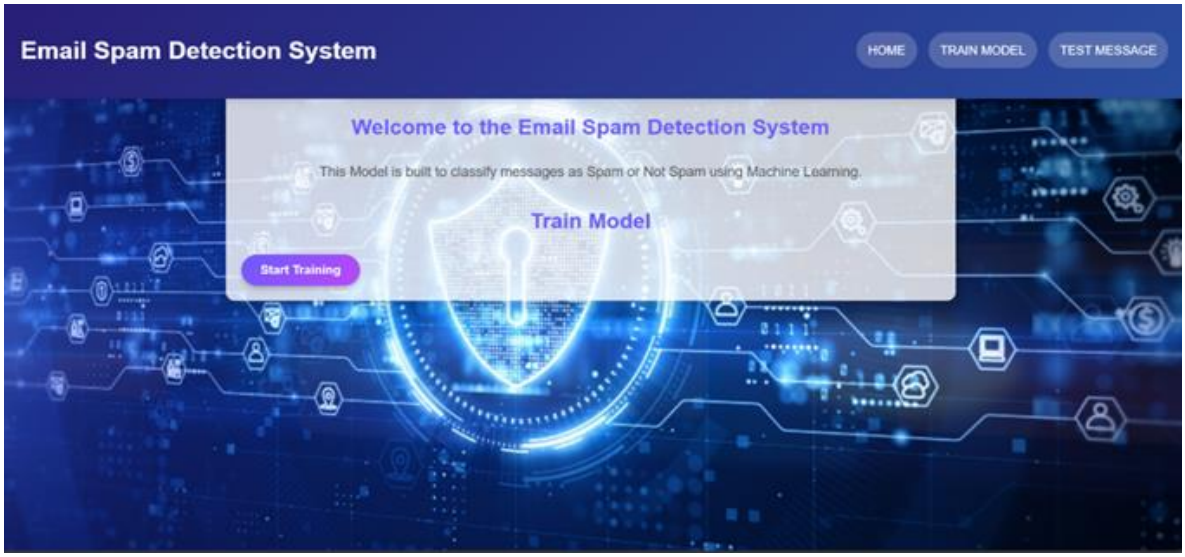

Figure 3: This snapshot displays the train model interface where the detection system is trained in such a way that it is able to classify spam and ham mails

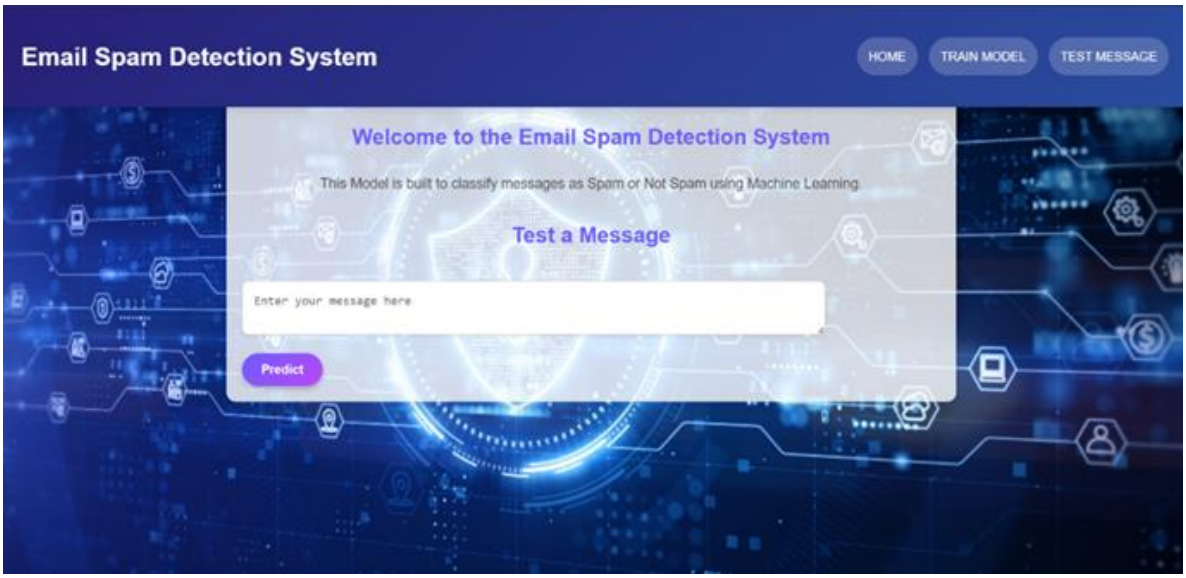

Figure 4: This snapshot displays test message which is used to which tests if the message is spam or not

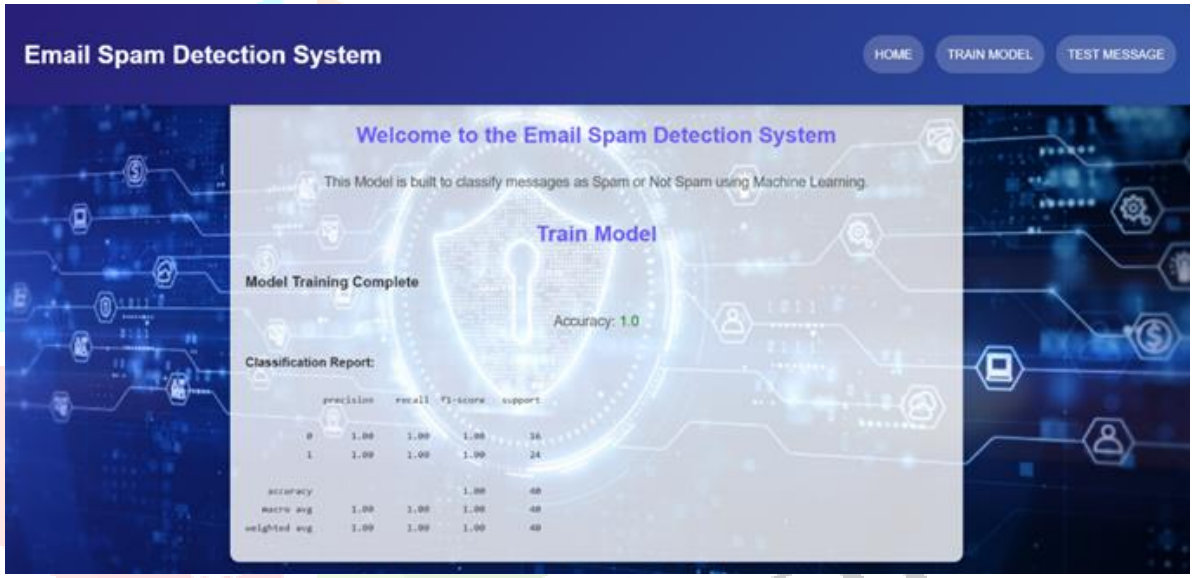

Figure 5: This snapshot gives us the result of the model after training

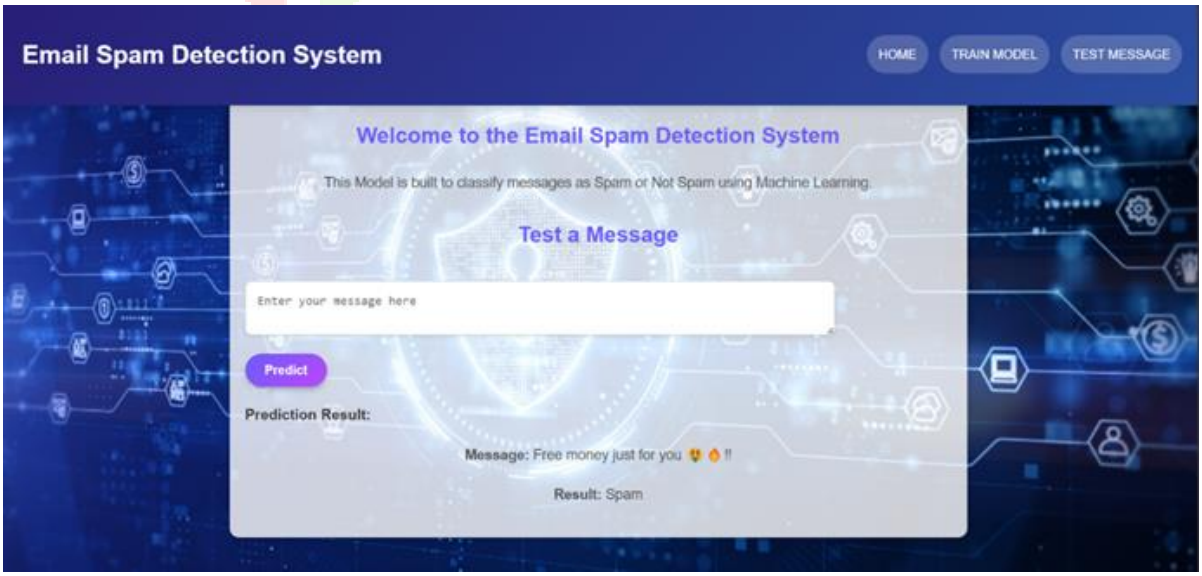

Figure 6: This snapshot shows the prediction result of a message which is SPAM

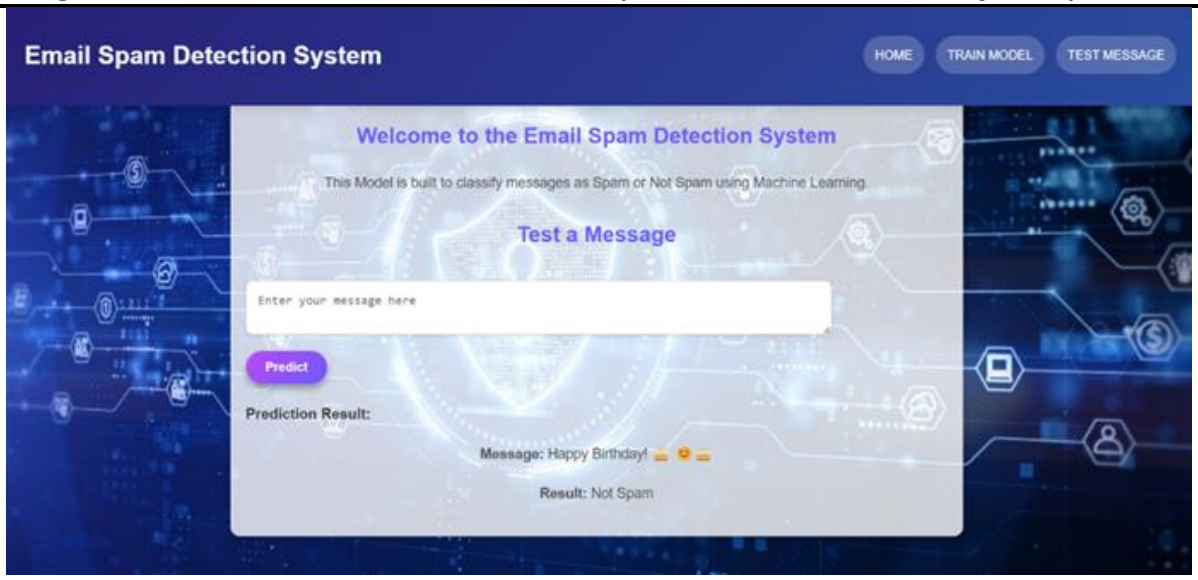

Figure 7: This snapshot shows the prediction result of a message which is HAM

VII. Conclusion and Future Scope

Email spam detection using machine learning effectively filters unwanted emails by analyzing text and emoji-based patterns. The system preprocesses data, extracts key features, and applies models like Random Forest to classify messages as spam or ham. With a user-friendly **Flask** interface, users can train models, test messages, and view performance metrics. The model's adaptability ensures improved accuracy with continuous training. This approach provides an efficient, scalable, and intuitive solution for managing email security.

The future of spam detection lies in **deep learning** models like CNNs and RNNs for improved accuracy. **Context-aware detection** can analyze sender behavior and email structure for better classification. **Integration with email clients** (Gmail, Outlook) will allow real-time filtering. **Personalized spam detection** can tailor results to user preferences, and **real-time learning** will ensure continuous updates against evolving threats. Addressing concerns like **false positives, privacy, scalability, and adversarial attacks** will further enhance spam detection systems.

VIII. REFERENCES

[1] V. Chandola, A. Banerjee, V. Kumar. Anomaly Detection in Network Traffic Using Supervised Machine Learning Ways, 2009

[2] S. Ahmed, H. Mahmoud, K. G. Srinivasa, M. H. J. Z. A Survey on Anomaly Detection in Network Traffic, 2016

[3] M. Hodge, S. K. Bansal. Network Anomaly Discovery with Random Cut Forest, 2017

[4] K. R. Chakrabarty, R. Krishnan, D. Srinivasan. A Machine Learning Approach for Network Traffic Classification and Anomaly Detection, 2018

[5] L. Chen, Z. Y. Zhao, H. Li. Deep Autoencoding Models for Network Anomaly Detection, 2019

[6] Tian, M. Liu, X. Wang, Z. Li. Deep Learning for Anomaly Detection in Network Traffic, 2020