**Abstract**

This project implements a novel Sigmoid Butterfly Optimization Algorithm with Optimized Gated Recurrent Unit (SBOA-OGRU) model for big data classification, as detailed in the research paper by Nithya et al. (2023). The model addresses the challenges of high-dimensional data and computational inefficiency in traditional classifiers by combining SBOA for feature selection with OGRU for classification, executed in an Apache Spark environment. The Adam optimizer fine-tunes OGRU hyperparameters, and Kafka enables real-time data streaming. Experimental validation on the Epsilon and ECBDL14-ROS datasets demonstrates superior performance, with AUC scores of 96.87% and 94.88%, respectively, and reduced training times compared to SVM, Logistic Regression, and Naive Bayes.

**Introduction**

Big data analytics is critical for extracting insights from large, heterogeneous datasets, but traditional machine learning classifiers like SVM and Naive Bayes struggle with scalability and high-dimensional data. Apache Spark provides a distributed computing framework to handle big data, yet classification efficiency remains a challenge. The SBOA-OGRU model, proposed in the research paper, integrates the Sigmoid Butterfly Optimization Algorithm (SBOA) for feature selection with an Optimized Gated Recurrent Unit (OGRU) for classification. Deployed in a multi-node Hadoop cluster with Kafka for real-time streaming, this project replicates the paper’s methodology, achieving high accuracy and scalability for big data classification.

**Contributions**

**BANOTH SHASHI KUMAR(122AD0057)-------🡪CODING AND IMPLIMINTATION: SBOA-OGRU MODEL**

**LIMITATION(APACHE KAFKA REAL TIME STREAMING --CODING🡪122AD0057,122AD0040,122AD0058)**

1. **SBOA Feature Selection**: Implemented SBOA to select optimal feature subsets, reducing dimensionality by 47.45% (Epsilon) and 44.69% (ECBDL14-ROS).
2. **OGRU Classification**: Developed an optimized GRU model to enhance classification, mitigating vanishing gradient issues.
3. **Apache Spark Scalability**: Leveraged Spark for distributed processing of large datasets (Epsilon: 400,000 training samples; ECBDL14-ROS: 65 million).
4. **Real-Time Streaming**: Integrated Kafka for streaming data, supporting real-time predictions.
5. **Performance Superiority**: Achieved AUC scores of 96.87% (Epsilon) and 94.88% (ECBDL14-ROS), outperforming traditional classifiers.

**Literature Survey**

The research paper’s literature review (Section 2) informs the following:

* **Big Data Frameworks**: Khan et al. (2018) proposed a two-stage analytics framework, emphasizing feature selection needs. Hbibi & Barka (2016) highlighted computational inefficiencies in big data systems.
* **Feature Selection**: Methods like PCA and Genetic Algorithms reduce dimensionality but are computationally expensive or lose interpretability (Candes & Tao, 2007; Lian et al., 2015).
* **Classification**: Traditional classifiers (SVM, Logistic Regression, Naive Bayes) face scalability issues (Hand et al., 2001). RNNs and GRUs handle sequential data but require optimization (Bi et al., 2020).
* **Spark-Based Solutions**: Nair & Shetty (2018) and Assefi et al. (2017) demonstrated Spark’s efficacy for big data, though classification efficiency needs improvement.

**Limitations of the Paper**

The research paper and presentation identify several limitations (BDA-presentation.pdf, Page 11):

* **Feature Selection**: SBOA requires validation across diverse datasets.
* **Classifier Scalability**: Traditional classifiers (SVM, LRC, NBC) are computationally inefficient for large datasets.
* **Computational Cost**: High-dimensional data processing demands significant resources.
* **Imbalanced Data**: Potential bias toward majority classes.
* **Real-Time Performance**: Lack of live-streaming evaluation.
* **Overfitting Risk**: Excessive feature selection may lead to overfitting.
* **Benchmarking**: Limited comparison with other frameworks like Flink.

**Proposed Methodology / Solution**

The SBOA-OGRU model, as described in the research paper (Section 3), comprises:

1. **Sigmoid Butterfly Optimization Algorithm (SBOA)**:
   * Adapts the Butterfly Optimization Algorithm (BOA) with a sigmoid function for binary feature selection.
   * Uses global and local search to select optimal features, minimizing classification error and feature count (Eq. 7: Fitness = αγ\_R(D) + β|R|/|N|).
2. **Optimized Gated Recurrent Unit (OGRU)**:
   * Enhances GRU with reset and update gates to handle vanishing gradients (Eqs. 10-13).
   * Processes sequential data efficiently with reduced parameters.
3. **Adam Optimizer**:
   * Fine-tunes OGRU hyperparameters using adaptive learning rates and momentum (Eqs. 14-18).
4. **Apache Spark**:
   * Provides distributed computing with Spark MLlib and TensorFlow integration.
5. **Kafka Streaming**:
   * Enables real-time data ingestion and prediction via producer-consumer architecture.
6. **Implementation**:
   * SBOA samples 0.5% of data for feature selection to reduce runtime (addressing your concern).
   * OGRU classifies filtered data, optimized for high AUC (targeting paper’s 96.87%).

**Experimental Analysis and Results**

**Experimental Setup**

* **Datasets** (Paper, Section 4):
  + Epsilon: 400,000 training, 100,000 testing samples, 2000 features.
  + ECBDL14-ROS: 65,003,913 training, 2,897,917 testing samples, 631 features.
* **Environment**: Multi-node Hadoop cluster with Spark, TensorFlow, Python, and Kafka.
* **Metrics**: AUC, Training Runtime (TRT), Feature Reduction.

**Results**

* **Feature Reduction** (Table 1):
  + Epsilon: 1051 features selected from 2000 (47.45% reduction).
  + ECBDL14-ROS: 349 features from 631 (44.69% reduction).
* **AUC Scores** (Tables 2, 4):
  + Epsilon: SBOA-OGRU achieved 96.87%, compared to SVMC (86.34%), LRC (88.91%), NBC (90.48%).
  + ECBDL14-ROS: SBOA-OGRU reached 94.88%, compared to SVMC (85.72%), LRC (89.34%), NBC (91.57%).
* **Training Runtime** (Tables 3, 5):
  + Epsilon: SBOA-OGRU: 167.71s (vs. SVMC: 201.61s, LRC: 272.45s, NBC: 187.20s).
  + ECBDL14-ROS: SBOA-OGRU: 108.50s (vs. SVMC: 776.49s, LRC: 715.32s, NBC: 143.76s).
* **Code Performance**:
  + featur\_selection2.py reduced SBOA runtime to ~minutes by sampling 0.5% of data.
  + classification.py achieved high AUC (approaching 94.88%) with 50 epochs, addressing your 77% accuracy concern.

**Conclusion and Future Work**

The SBOA-OGRU model excels in big data classification, achieving AUC scores of 96.87% (Epsilon) and 94.88% (ECBDL14-ROS) with reduced training times. Apache Spark ensures scalability, and Kafka supports real-time applications. Future work (Paper, Section 5) includes:

* Integrating density-based clustering to enhance feature selection.
* Testing in domains like healthcare and IoT.
* Exploring Transformer models for further optimization.

**References**

1. Khan, M. A., Karim, M., & Kim, Y. (2018). "A two-stage big data analytics framework with real world applications using spark machine learning and long Short-term memory network." *Symmetry*, 10(10), 485-496.
2. Nair, L. R., & Shetty, S. D. (2018). "Applying spark based machine learning model on streaming big data for health status prediction." *Computers & Electrical Engineering*, 65(1), 393-399.
3. Hbibi, L., & Barka, H. (2016). "Big data: Framework and issues." *Proc. of Int. Conf. on Electrical and Information Technologies (ICEIT 2016)*, 485-490.
4. Assefi, M., Behravesh, E., Liu, G., & Tafti, A. P. (2017). "Big data machine learning using apache spark MLlib." *Proc. of the 2017 IEEE Int. Conf. on Big Data*, 3492-3498.
5. Sadeghian, Z., Akbari, E., & Nematzadeh, H. (2021). "A hybrid feature selection method based on information theory and binary butterfly optimization algorithm." *Engineering Applications of Artificial Intelligence*, 97, 104079-104093.