Jurnal BPPK Volume 16 Nomor 1 Tahun 2023 Halaman 67-77

BADAN PENDIDIKAN DAN
PELATIHAN KEUANGAN
KEMENTERIAN KEUANGAN
REPUBLIK INDONESIA

JURNAL BPPK

CUSTOMS FRAUD DETECTION USING EXTREMELY BOOSTED NEURAL NETWORK (XBNET)

Adrianta Ras Sembiring[a], Diyouva Christa Novith[b], Muhammad Hadiyan Ridho[c]

[a] Directorate General of Customs and Excise, Indonesia. Email: adrianta.ras@customs.go.id

[b] Directorate General of Customs and Excise, Indonesia. Email: diyouva.christa@customs.go.id (penulis berkorespondensi)

[c] Directorate General of Customs and Excise, Indonesia. Email: hadiyan.ridho@customs.go.id

ARTICLE INFORMATION

ARTICLE HISTORY
Received
3 October 2023

Accepted to be published
24 November 2023

KEYWORDS:
Customs
Fraud Detection
Import
International Trade
XBNet

ABSTRACT

Pentingnya peran perdagangan lintas batas membuat Bea Cukai berperan penting tidak hanya dalam menjaga rantai pasokan, namun juga dalam mengamankan pendapatan negara dari praktek kecurangan. Besarnya volume perdagangan internasional dan terbatasnya tenaga kerja membuat Bea Cukai di seluruh dunia harus menerapkan manajemen risiko yang efisien dan efektif. Penelitian ini mengusulkan XBNet, kumpulan algoritma berbasis pohon dengan algoritma pembelajaran mendalam, untuk mendeteksi penipuan dalam aktivitas impor. Keunggulan XBNet adalah menggabungkan pohon yang ditingkatkan gradien dengan jaringan saraf di mana bobot, bias, dan kerugian disesuaikan secara bersamaan dengan fitur penting setiap pohon di setiap lapisan. Objek penelitian ini adalah data Pemberitahuan Impor dari empat Kantor Pabean, dan model yang ditetapkan pada masing-masing Kantor Pabean untuk menangkap pola penipuan terkait wilayahnya. Kami membandingkan model dengan dua parameter berbeda dan menyimpulkan bahwa model dengan *learning rate* = 1%, jumlah *hidden layer* = 2, *activation function* = sigmoid, dan jumlah *epoch* = 100 adalah yang paling cocok untuk Belawan, Merak, dan Makassar dan model dengan jumlah *hidden layer* = 2, jumlah *epoch* = 50 dan parameter lainnya diatur sama dengan yang paling sesuai untuk Tanjung Emas.

Noticing the vital role of cross-border trade has made Customs plays a crucial role not only in maintaining supply chain, but also in securing government revenue from intentional fraud. Given the huge volume of international trade and limited workforce, Customs across the world must implement efficient and effective risk management. This paper proposes XBNet, an ensemble of tree-based algorithms with deep learning algorithms, to detect fraud in import activity. The strength of XBNet is combining gradient-boosted trees with neural networks where the weights, bias, and loss are adjusted simultaneously with the importance features of each tree in each layer. The object of this study is Import Declaration data from four Customs Offices, and the model is set for each Customs office to capture fraud patterns related to their region. We compared the model with two different parameters and concluded the models with learning rates = 1%, number of hidden layers = 2, activation function = sigmoid, and number of epochs = 100 as the most suitable for the Belawan, Merak, and Makassar and model with number of hidden layers = 2, number of epochs = 50 and other parameters are set the same as the most suitable for Tanjung Emas.

1. INTRODUCTION

The post-pandemic era has made the world aware of how important it is to maintain supply chain sustainability, especially for essential commodities whose domestic production resources are limited (Novith & Ridho, 2020).

It makes cross-border trade hold a vital role and customs authorities be in charge of huge volumes of transactions. As cross-border trade traffic is growing rapidly, as is emerging fraudulent trade. Market players tend to manipulate their customs declaration intentionally to evade tax or reduce the paid duty. Moreover, World Customs Organization (2020) reported the escalation in numbers of attempted fraud and tax evasion occurrence. In order to make sure the stakeholders fulfilled their responsibilities, customs have to conduct physical inspection.

Physical inspection is the only traditional way to check the declared documents and the imported goods are matched and follow applicable regulations. But doing physical inspection for all imported goods is arduous and costly. According to the World Bank, the ratio of conducting physical inspection of 72% of countries across the world was less than 30% whilst in Indonesia, there is only 8% of import shipments that have been physically inspected (Arvis et al., 2018). To minimize the negative impact that is possible to deliver due to lack of importer goods supervision resulting from low inspection ratios, customs must implement risk management effectively to detect illicit trade, mainly for import activities.

Detecting fraud commonly has been conducted by employing rule-based algorithms, which will change to machine learning algorithms gradually (Mikuriya & Cantens, 2020). Deploying rule-based algorithms would be a burden if the country has limited capable officers that are qualified to construct the rules manually and curate a rule set periodically. In fact, many developing countries are in lack of capable officers (Park et al., 2022). Moreover, replacing one to another rule set must be going through bureaucracy. This condition will make the given rules less agile to adapt to the rapid changes in technology as well as sophisticated crime techniques.

In order to manage an astronomical amount of transactions with limited manpower, it necessitates automation of fraud detection using machine learning algorithms. Employing machine learning using various algorithms at national level has been done many times (Vanhoeyveld et al., 2020) and many countries are requesting assistance to adapt machine learning for detecting fraud. Unfortunately, even though customs actively support building and implementing machine learning into their system, the application in the system has been a limited task due to the proprietary nature of the data, so the adoption is slow in the customs field (Jeong et al., 2022). Moreover, adapting machine learning algorithms remains several challenges, such as interpretability, availability of labeled data, imbalanced dataset constraint, and privacy concerns.

Based on the preliminary explanation mentioned above, we aim to make customs fraud detection using machine learning algorithms from import declaration in Indonesia. Our study focuses on four customs offices and determines to craft models for each customs office in regard to capturing fraud patterns in different regions.

This study employs the Extremely Boosted Neural Network (XBNet) model, which is a novel model that is an ensemble of tree-based algorithms with deep learning algorithms. Hopefully, this study can contribute to enrich the development of machine learning in order to detect import declaration fraud.

2. LITERATURE REVIEW

2.1. Customs Fraud Detection

There are various ways that customs authorities across the world undergo to identify fraud regarding their authority. The earlier attempts on detecting fraud by using rule-based systems and random selection algorithms. Rule-based methods need the knowledge of domain experts and be codified (Butgereit, 2021). Even though it is straightforward and interpretable, but prone to an ever-changing pattern over time. Moreover, a rule-based system is cumbersome to develop, maintain and subjective to expert knowledge (Pozzolo et al., 2017). Regardless of the limitation, most customs administrations are still using rule-based algorithms (Kültür & Çağlayan, 2017).

Furthermore, many customs across countries have already implemented machine learning (Singh et al., 2023). In Indonesia, we used to depend on a profiling matrix to classify the level of risk on imported goods, then by the end of 2019, Indonesia employs a risk engine based on machine learning algorithms. However, each machine learning algorithm has its limitations, such as limited labels of data and unavailability of data and the limitation of its model. In this case, we propose Extremely Boosted Neural Network (XBNet) model.

2.2. Tree Based Model and Neural Networks

Data-driven approaches using machine learning become very important in many areas (Linardatos et al., 2021). Nonetheless, the widespread use of machine learning algorithms makes the interpretability of predictive models vital, mainly in terms as ethics are involved, for instance, medicine, law and finance (Yang et al., 2018). Many data analysts attempt to overcome the weakness by using an ensemble method of machine learning (Mai et al., 2021).

Ensemble machine learning is a general approach to seek a better performance of machine learning by combining multiple models of algorithms (Kim et al., 2022; Kim et al., 2020; Yang et al., 2018). Using a single model is not sufficient to meet the need for solving complex and various problems due to each model has its own disadvantages and advantages (Thongsuwan et al., 2021), by developing a combined model we can get better results and avoid the limitations. Thus, we proposed XBNet that combines neural networks and tree-based models to craft a robust architecture to enhance model's performance and interpretability. We employ an optimization technique, Boosted Gradient Descent, as an extension of any gradient-based optimization technique using tree-based models (Ruder, 2016). In this method, each layer will train trees to extract feature importance, which is used along with the weights determined by gradient descent for adjusting the weights of those layers respectively.

3. RESEARCH METHODOLOGY

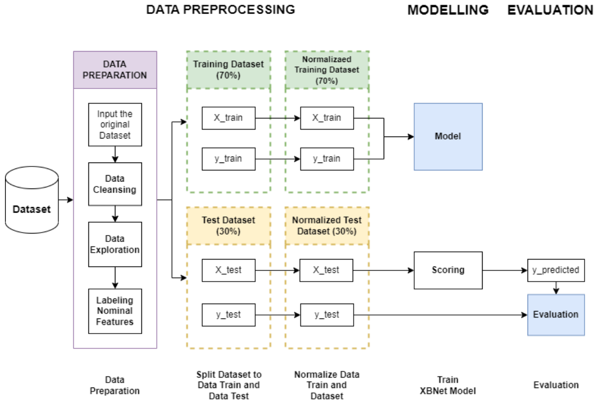
Bagian ini menguraikan metode seleksi dan pengumpulan data, pengukuran dan definisi operasional variable, dan metode analisis data. Sama seperti bagian ke dua di atas, sejak paragraph pertama, kalimat pertamanya juga ditulis menjorok ke dalam satu tabulasi.

3.1. Data Analytic Framework

This analysis is conducted according to a scientific method which consists of three steps, namely: Data Preprocessing, Modelling, and Evaluation. Overall, the framework of this analysis can be seen at Figure.1.

Data analytic framework is useful as a guidance practice for the analyst to keep the analytical process still systematic and to assure the data analytics project can achieve its objective. Besides that, the framework will help the reviewer or peer to understand the analytics project.

Figure.1 Data Analytic Framework



3.1.1. Datasets

This analysis uses transaction-level level data of Indonesia Import Declaration (PIB). The data pose import declarations from 4 Customs Offices namely: Tanjung Emas, Belawan, Merak, and Makassar from January to June 2022. Table.1 shows the number of data populations that are used in this analysis. The data collected are historical import declaration data that has been submitted by the importers and officaly inspected and approved by the customs documents inspector (PFPD). The illicit rate shows the percentage of data of fraud in import declaration.

Table.1 Data Population

Customs Office	Total Data	Illicit Rate
Tanjung Emas	532,748 records	7.3%
Belawan	158,220 records	5.1%
Merak	6,554 records	1.1%
Makassar	4,152 records	0.4%

Source : processed by authors

Table.2 shows the list of data fields that will be used as features in the model. The dataset consists of 26 features and can be divided into 3 categories namely Customs Declaration Features (17 features), Risk Label Features (8 features), and Prediction Target (1 features).

To conduct our analysis, the data must be able to be identified whether it has fraud or not. The illicit features, as prediction target, indicate whether the import transaction has fraud or no fraud. A binary flag is used to identify the fraud; 0 if the transaction has no fraud and 1 if the transaction is detected to be fraudulent.

3.1.2. Handling missing value

In deep learning models, missing value in the dataset must be treated (handled) in order to avoid the model leading to wrong prediction. For example, in our dataset, there are missing value records in the customs broker, and this information indicates that there are not involved customs broker customs clearance processes. We replace the missing (NaN) value with a certain code.

3.1.3. Data splitting

Dataset is split into training dataset and testing dataset. Splitting is set based on percentage ratio : 70% of the dataset is used for training the model and 30% of the dataset is used as a testing set.

3.1.4. Label encoding

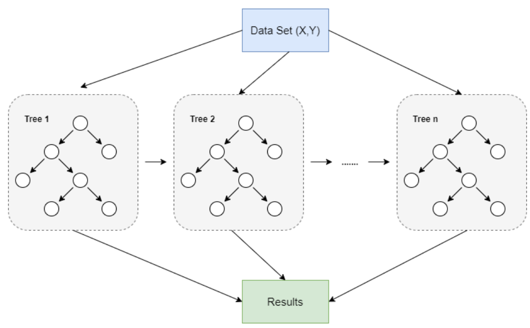
All features in Table.2 will be used as input in our model. To ensure that the model can run properly, we transform all nominal values into continuous values in the forms of number code. The features named with '_LABEL'in Table.2 are the features which are encoded.

3.2. Model: XBNet

3.2.1. Concept of XGBoost

XGBoost is an ensemble tree-based model that uses a boosting method to increase prediction performance results. In the XGBoost model, the prediction process is carried out by growing the decision tree in stages and sequentially to get better prediction results. Figure.2 shows the structure of XGBoost in general.

Figure.2 Structure of XGBoost



The prediction of XGBoost is made by aggregating the tree results in the following way:

$$y = \psi(x) = \sum_{n=1}^{N} g_n(x)$$

Where x is the input features, y is the outputs, and gn(x) represents the score of the leaf of the Nth tree.

XGBoost algorithm is an advanced method by using gradient boosting and can be used both for classification and regression. The algorithm uses an ensemble model based on a set of machine learning classifiers called weak learners (Ouadjer et al., 2020).

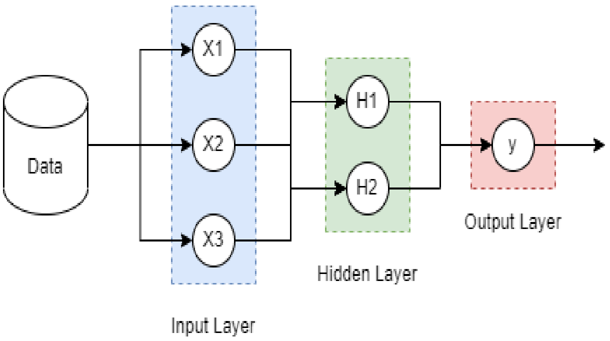
Gradient Boosting. When a decision tree is a weak learner, the resulting algorithm is called gradient boosted trees. Gradient boosting starts by generating an initial classification tree and continues to fine-tune new trees through loss function. In practical terms, the gradient boosting regression model provides many tunable hyperparameter and loss functions, which make it more flexible and accurate compared to the other three models (Cai et al., 2020).

Features Rank. The XGBoost model can show what the most important feature in the model is by making a rank of features that are used in the input model.

3.2.2. Concept of Neural Networks

Neural Networks are a main foundation of deep learning algorithms. It is inspired by how the human brain processes information. Neural networks consist of 3 important parts, namely : Input Layer, Hidden Layer, and Output Layer. Figure.3 shows the structure and the process of Neural Network.

Figure.3 Neural Network Structure and Process



Forward Propagation. This is the first step in Neural Networks. In this step, data is sent from input layer to output layer through the networks. Each neuron receives the input and then multiplies with the weight number and adds with bias. Next, the input data will be forwarded to the activation function.

Activation Function - Sigmoid. Activation function has a task to turn the input into output. Sigmoid function receives any number and changes the input into a ranging number between 0 and 1.

Backward Propagation and weight adjustment. After the forward propagation process, the neural network will produce an output. The output will compare with the real output to calculate loss computation. The loss computation will be backward propagated and adjusted according to the loss value. This process will happen repeatedly until getting the accepted minimum value.

Gradient Descent. Gradient descent uses mathematical calculation to find the minimum value of cost function (loss). In each stage, the weight and bias are adjusted by decreasing the cost with gradient and multiplying with learning rate.

3.2.3. Concept of XBNet

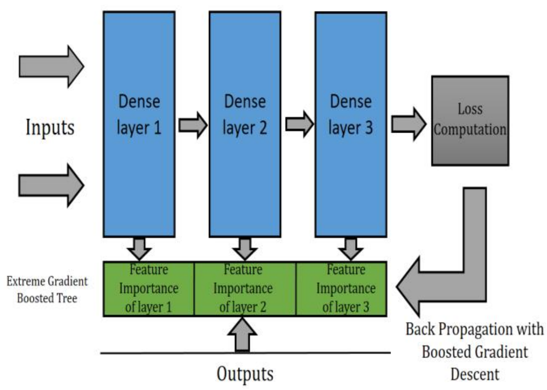
XBNet is a novel method that was developed and introduced by Tushar Sarkar in 2022. Nowadays, boosted-trees are the most popular method to make machine learning for tabular data, but on the other side, neural networks method has been proven to process unstructured data like images, text, video, and audio (Sarkar, 2022).

Through several experiments, neural networks can result a model with high accuracy but this algorithm has a weakness. The characteristic of deep learning is that it is difficult to find what happened in the model building and what are the most important features that build a model and its accuracy. Moreover, deep learning is difficult to translate in order to implement.

On the other hand, a tree-based model is more transparent and easy to implement. Furthermore, the gradient boosting helps to avoid overfitting because it takes advantage of a lot of decision trees in stages and sequentially to get better prediction results.

Observing the strengths and weaknesses of each method, the idea of XBNet is trying to combine boosted-trees method and neural networks method. Overall, the architecture of XBNet can be seen at Figure.4

Figure.4 XBNet Architecture



Source: Sarkar, 2022

XBNet architecture makes a sequential structure of hidden layers for all input and output data. Each layer is connected to gradient boosted tree which is separated from the architecture. Trees are trained in every layer of the architecture and the features importance of each tree is used to determine the adjusted weights of those layers by gradient descent.

When the training model is run, the input data is processed through forward and backward propagation, and the weights and bias of all layers get adjusted according to gradient descent.

Entering the next training epoch, the data through all the layers again and simultaneously adjust the weights based on the feature importance of the gradient boosted tree which is trained on each layer.

3.3. Evaluation

To evaluate the performance of the model, we use 3 metrics:

1) Precision - This metric measures how many of the transaction data have been successfully predicted among the whole data.
2) Recall - This metric measures how many of the transaction data have been successfully predicted.
3) F1-Score - This metric is the combination of precision and recall, which measures model accuracy on a dataset.

4. RESULTS AND FINDINGS

In this analysis, the data collected from four Customs Offices in Indonesia from different regions. The aim is to make a specified model for each custom office, so the training and testing dataset is run based on the data from each office.

We set the parameters of model with number of input features, number of output features, percentage of learning rate, number of hidden layers, activation function, and number of epochs. To run the neural networks algorithm, we need 2 important parameters namely the number of hidden layers and number of epochs. There is no specific formula to define how much the number is, so we started with a random number for the hidden layer and epochs. As a default parameter, we set the parameter with number of input features = 25, number of output features = 25, learning rates = 1%, number of hidden layers = 2, activation function = sigmoid, and number of epochs = 100.

With this default parameters, we found that when model with 2 layers and number of epochs is 100, it yields 72.72% validation accuracy and 2.32% validation loss in Tanjung Emas, 97.12% validation accuracy and 2.29% validation loss in Belawan, 99.23% validation accuracy and 2.29% validation loss in Merak, and 98.71% validation accuracy and 2.28% validation loss in Makassar.

4.1. Hyperparameter Tuning

We evaluated our model on customs fraud detection by creating 2 other models with different parameters. The number of hidden layers and number of epochs are used in parameters tuning because these are the most important parameters.

Next, the second model with 2 layers and number of epochs is 50 and other parameters stay the same, it yields 77.03% validation accuracy and 2.31% validation loss in Tanjung Emas, 95,48% validation accuracy and 2.30% validation loss in Belawan, 98.93% validation accuracy and 2.29% validation loss in Merak, and 98.63% validation accuracy and 2.28% validation loss in Makassar.

Finally, the third model with 3 layers and number of epochs is 200 and other parameters stay the same, it yields 77.03% validation accuracy and 2.31% validation loss in Tanjung Emas, 97.12% validation accuracy and 2.29% validation loss in Belawan, 99.23% validation accuracy and 2.29% validation loss in Merak, and 98.71% validation accuracy and 2.28% validation loss in Makassar. The comparison of validation accuracy and validation loss after tuning the parameters of the model can be seen in Table 3.

Table.3 Validation Accuracy and Loss Comparison After Parameters Tuning

Customs Office Accuracy and Loss Validation	Parameters		
	lr=1%, hl=2, e=50	lr=1%, hl=2, e=100	lr=1%, hl=3, e=200
Tanjung Emas			
• Acc	**77.03**	72.72	36.84
• Loss	**2.31**	2.32	2.33
Belawan			
• Acc	95.48	**97.12**	85.70
• Loss	2.30	**2.29**	2.40
Merak			
• Acc	98.93	**99.23**	98.01
• Loss	2.29	**2.29**	2.29
Makassar			
• Acc	98.63	**98.71**	98.55
• Loss	2.28	**2.28**	2.28

*Acc: validation accuracy; *Loss : validation loss;
*lr : learning rates; *hl : hidden layers; *e : epochs

Source : processed by authors

As shown in Table.3, instead of increasing the accuracy of the performance of the model, increasing the number of hidden layers and epochs actually reduces the score of validation accuracy and increases the score of validation loss. This happened in all Customs Office models after the completion of the training dataset.

Increasing the number of hidden layers much more than the sufficient number that can be set will lead to a decreasing in accuracy, and this can be seen as a warning that your network will lead to overfitting. After a certain point of the data training process, increasing the number of epochs also can lead to overfitting, because the model takes more memory and performs poorly against new and unseen patterns. It will affect the model's performance and experience a drop in accuracy. Exceeding the number of hidden layers and epochs will no longer increase the accuracy score (Li et al., 2019).

4.2. Model selection

In deep learning, increasing the number and hidden layer and number of epochs doesn't always go concurrently with the result. In the Tanjung Emas Office model, the score of accuracy increases when the number of epochs is reduced. There are many factors that can affect the result such as the number of the dataset, the complexity of the dataset, architecture of model, and parameters setting. Choosing the optimum formula of the model carefully is needed to make a better model selection.

Moreover, analyzing the testing performance metric will help to consider in selecting the best model. Table 4, Table 5, and Table 6 show the performance metric from each model. Overall, based on the performance metric, the model from each customs office can predict accurately whether the import declaration has fraud (class 1) or has not (class 0).

Table.4 Testing Performance Metric of the 1st model with Hidden Layer = 2 and Epochs = 100

Customs Office and Class	Metric		
	Pre	Rec	F-1
Tanjung Emas			
• Class 0	0.58	0.98	0.73
• Class 1	0.98	0.58	0.73
Belawan			
• Class 0	**0.96**	**1.00**	**0.98**
• Class 1	**1.00**	**0.87**	**0.93**
Merak			
• Class 0	**1.00**	**1.00**	**1.00**
• Class 1	**0.74**	**0.81**	**0.77**
Makassar			
• Class 0	**0.99**	**1.00**	**0.99**
• Class 1	**1.00**	**0.11**	**0.20**

*Pre : precision, *Rec : Recall, *F-1 : F-1 Score

Source : processed by authors

Table.5 Testing Performance Metrics Of the 2nd Model with Hidden Layer = 2 and Epochs = 50

Customs Office and Class	Metric		
	Pre	Rec	F-1
Tanjung Emas			
• Class 0	**0.62**	**0.98**	**0.76**
• Class 1	**0.98**	**0.65**	**0.78**
Belawan			
• Class 0	0.95	1.00	0.97
• Class 1	1.00	0.79	0.88
Merak			
• Class 0	1.00	0.99	0.99
• Class 1	0.62	0.84	0.71
Makassar			
• Class 0	0.99	1.00	0.99
• Class 1	1.00	0.06	0.11

*Pre : precision, *Rec : Recall, *F-1 : F-1 Score

Source : processed by authors

Table.6 Testing Performance Metrics Of the 3rd Model with Hidden Layer = 3 and Epochs = 200

Customs Office and Class	Metric		
	Pre	Rec	F-1
Tanjung Emas			
• Class 0	0.37	1.00	0.54
• Class 1	0.00	0.00	0.00
Belawan			
• Class 0	0.85	0.99	0.92
• Class 1	0.94	0.35	0.51
Merak			
• Class 0	0.98	1.00	0.99
• Class 1	0.10	0.03	0.05
Makassar			
• Class 0	0.99	1.00	0.99
• Class 1	0.00	0.00	0.00

*Pre : precision, *Rec : Recall, *F-1 : F-1 Score

Source : processed by authors

There is a problem in the model of Makassar where the model is indicated to predict the fraud (class 1) inaccurately because the Recall and F1 score of class 1 are too low while the accuracy is high. The cause of this problem can be resulted by the imbalanced class of the datasets.

After analyzing the result of hyperparameters tuning and performance metric, we choose the first model with learning rates = 1%, number of hidden layers = 2, activation function = sigmoid, and number of epochs = 100 as the best model for Belawan, Merak, and Makassar and the second model with learning rates = 1%, number of hidden layers = 2, activation function = sigmoid, and number of epochs = 50 as the best model for Tanjung Emas.

Figure.5 (Appendix) depicts the trend line of accuracy and loss score after finishing a number of epochs of data training process. Overall, Belawan model has a better trend of accuracy and loss rather than other model, because there is an increasing trend in accuracy accompanied by decreasing trend of loss although the trend fluctuated until the last epoch. In the Merak and Makassar model, there are 'step/static' trend in accuracy and loss, but it is normal when the accuracy trend doesn't show a decreasing trend. On the other hand, if there is an increasing trend of loss when the accuracy trend is static, it is indicated to be overfitting.

4.3. Features Importance

A benefit using gradient boost is retrieving the importance scores for each feature after boosted trees are constructed. Features importance will be the interest point to understand the trained predictive model. Moreover, importance can be seen as the amount of contribution to trained model performance. Figure.6 (Appendix) depict the comparison of features importance of each selected model.

5. CONCLUSIONS

The growth of international trade flows have made the customs administration craft effective systems in order to detect suspicious transactions. Currently, the use of machine learning algorithms can be integrated with customs administration systems to help the detection of suspicious transactions.

XBNet is a machine learning model that combines neural networks with gradient boosted trees. This study has presented a novel machine learning method by using XBNet method to detect (predict) the indication of customs fraud in import declaration.

Making a good model for customs fraud detection is a challenge faced by Indonesia Customs.

The XBnet model presented in this study can be used to predict suspicious transactions, and our model performance is good enough to predict suspicious or non-suspicious transactions accurately in the office with high or low volume of transactions in Tanjung Perak, Belawan, and Merak. But the trained model can't accurately predict the suspicious transactions in Makassar.

6. LIMITATION AND RECOMMENDATION

Bagian ini menjelaskan implikasi temuan dan keterbatasan riset, serta jika perlu memberikan saran yang dikemukakan Peneliti untuk riset-riset yang akan datang. Sama seperti bagian ke dua di atas, sejak paragraph pertama, kalimat pertamanya juga ditulis menjorok ke dalam satu tabulasi.

6.1. Limitations

This analysis tried to develop robust and effective machine learning to predict fraud in import transactions. However, there were some limitations in this analysis:

1) The weakness of deep learning algorithms required more time and resources for training gradient boosted trees in every layer. Due to the limitation of time and the hardware we used, the hyperparameters tuning used in the evaluation process only uses the number of hidden layers and number of epochs.
2) This analysis only uses 25 features namely as shown in Table.2 with 6-month periods as an input of the model.

6.2. Recommendations

According to the discussion and the limitations of this analysis, we give a recommendation for future research:

1) Creating several models with variations of parameter tuning to improve the quality of evaluation. For example, change the learning rate and activation function. Moreover, we must be careful that our model will not be overfitting or underfitting as a result of parameters tuning.
2) Exploring more variables as features for the input model and increasing the time length of the import declaration period. For example, using a combination of features and adding the profile of importer or commodity can be considered as additional features and increase the time period of import declaration to 12 months.

REFERENCES

Arvis, J.-F., Ojala, L., Wiederer, C., Shepherd, B., Raj, A., Dairabayeva, K., & Kiiski, T. (2018). *Connecting to Compete 2018: Trade Logistics in the Global Economy*. World Bank. http://hdl.handle.net/10986/29971

Butgereit, L. (2021). Anti Money Laundering: Rule-Based Methods to Identify Funnel Accounts. *Conference on Information Communications Technology and Society*. doi:10.1109/ictas50802.2021.9394990

Cai, J., Xu, K., Zhu, Y., Hu, F., & Li, L. (2020). Prediction and Analysis of Net Ecosystem Carbon Exchange Based on Gradient Boosting Regression and Random Forest. *Applied Energy*, *262*.

Jeong, C., Kim, S., Park, J., & Choi, Y. (2022). Customs Import Declaration Dataset. *10.48550/arXiv.2208.02484*. https://doi.org/10.48550/arXiv.2208.02484

Kim, S., Mai, T.-D., Han, S., Park, S., D.K, T. N., So, J., Singh, K., & Cha, M. (2022). Active Learning for Human-in-the-Loop Customs Inspection. *IEEE Transactions on Knowledge and Data Engineering*. 10.1109/TKDE.2022.3144299

Kim, S., Tsai, Y.-C., Singh, K., Choi, Y., Ibok, E., Li, C.-T., & Cha, M. (2020). DATE: Dual Attentive Tree-aware Embedding for Customs Fraud Detection. *Applied Data Science Track Paper*.

Kültür, Y., & Çağlayan, M. U. (2017). Hybrid Approaches for Detecting Credit Card Fraud. *Expert System*, *34*(2).

Li, H., Li, J., Guan, X., Liang, B., Lai, Y., & Luo, X. (2019). Research on Overfitting of Deep Learning. *15th International Conference on Computational Intelligence and Security (CIS)*.

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2021). Explainable AI: A Review of Machine Learning Interpretability Methods. *Theory and Applications of Information Theoretic Machine Learning*, *23*(1). https://doi.org/10.3390/e23010018

Mai, T.-D., Hoang, K., Baigutanova, A., & Alina, G. (2021). Customs Fraud Detection in the Presence of Concept Drift. *2021 International Conference on Data Mining Workshops (ICDMW)*, 370-379. 10.1109/ICDMW53433.2021.00052

Mikuriya, K., & Cantens, T. (2020). If Algorithms Dream of Customs, Do Customs Officials Dream of Algorithms? A Manifesto for Data Mobilisation in Customs. *Word Customs Journal*, *14*(2), 3-22. https://www.wcoomd.org/-/media/wco/public/global/pdf/topics/research/research-paper-series/48_manifesto_for_data_mobilization_for_customs.pdf?la=fr

Novith, D. C., & Ridho, M. H. (2020). Customs' Role in Overcoming Covid-19 Outbreak: The Case of Indonesia. *Customs Research and Application Journal*, *2*(2), 121-140. https://doi.org/10.31092/craj.v2i2.67

Ouadjer, Y., Adnane, M., & Bouadjenek, N. (2020). Feature Importance Evaluation of Smartphone Touch Gestures of Biometric Authentication. *2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, 103-107. doi: 10.1109/IHSH51661.2021.9378750

Park, S., Kim, S., & Cha, M. (2022). Knowledge Sharing via Domain Adaptation in Customs Fraud Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(11), 12062-12070. https://doi.org/10.48550/arXiv.2201.06759

Pozzolo, A. D., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2017). Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 1-14. DOI:10.1109/TNNLS.2017.2736643

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Sarkar, T. (2022). XBNet: An Extremely Boosted Neural Network. *Intelligent Systems with Applications*, *15*. XBNet: An Extremely Boosted Neural Network

Singh, K., Tsai, Y.-C., Li, C.-T., Cha, M., & Lin, S.-D. (2023). GraphFC: Customs Fraud Detection with Label Scarcity. *arXiv preprint arXiv*. https://doi.org/10.48550/arXiv.2305.11377

Thongsuwan, S., Jaiyen, S., Padcharoen, A., & Agarwal, P. (2021). ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost. *Nuclear Engineering and Technology*, *53*, 522-531.

Vanhoeyveld, J., Martens, D., & Peeters, B. (2020). Customs Fraud Detection - Assessing the Value of Behavioural and High Cardinality Data under the Imbalanced Learning Issue. *Pattern Analysis and Applications*, *23*, 1457-1477. https://doi.org/10.1007/S10044-019-00852-W

World Customs Organization. (2020, March 23). *COVID-19 Urgent Notice: counterfeit medical supplies and introduction of export controls on personal protective equipment*. Retrieved September 28, 2023, from https://www.wcoomd.org/en/media/newsroom/2020/march/covid_19-urgent-notice-counterfeit-medical-supplies.aspx

Yang, Y., Morillo, I. G., & Hospedales, T. M. (2018). Deep Neural Decision Trees. *arXiv preprint arXiv:1806.06988*. https://doi.org/10.48550/arXiv.1806.06988

APPENDIX

Type	Variables	Description
Customs Declaration Features	SUBMISSION_MONTH	The month in which the transaction occurred.
	IMPORTER_ID_LABEL	A code to identify an importer
	CUSTOMS_BROKER_ID_LABEL	A code to identify a customs broker
	COUNTRY_LABEL	A code to identify the origin of country
	IMPORT_FACILITY_CODE_LABEL	A code to identify the import trade facilitation
	FTA_FLAG_LABEL	A code to identify whether the importation use FTA tariff schema
	LOADING_PORT_LABEL	A code to identify the loading port
	TRANSIT_PORT_LABEL	A code to identify the transit port
	TARIFF_CODE	An 8-digit code indicating the applicable tariff of the item based on the harmonized system (HS).
	QUANTITY	The number of items in importation
	CONTAINER	The number of containers in importation
	TEUS	The number of 20 feet containers equivalent units
	NETTO	The weight of items in importation
	CURRENCY_LABEL	A code to identify currency of transaction
	VOYAGE_FLAG_LABEL	A code to identify flag of the carrier
	CIF	The value of the transaction, including the insurance and freight
	DECLARANT_TAX	The value of tax paid in customs declaration
Risk Tag Features	IMPORTER_RISK_LABEL	A code to identify whether the importers are risky or non-risky
	CUSTOMS_BROKER_RISK_LABEL	A code to identify whether the customs brokers are risky or non-risky
	COUNTRY_RISK_LABEL	A code to identify whether the countries are risky or non-risky
	TARCODE8_RISK_LABEL	A code to identify whether the 8-digit Tariff Code are risky or non-risky
	TARCODE6_RISK_LABEL	A code to identify whether the 6-digit Tariff Code are risky or non-risky
	TARCODE4_RISK_LABEL	A code to identify whether the 4-digit Tariff Code are risky or non-risky
	TARCODE2_RISK_LABEL	A code to identify whether the 2-digit Tariff Code are risky or non-risky
	TARCODE8_COUNTRY_RISK_LABEL	A code to identify whether the combination of 8-digit Tariff Code and countries are risky or non-risky
Prediction Target	ILLICIT	A binary flag to indicate whether the transactions have a fraud or no fraud

Source : processed by authors

Figure 5 Comparison of Accuracy and Loss Trendline

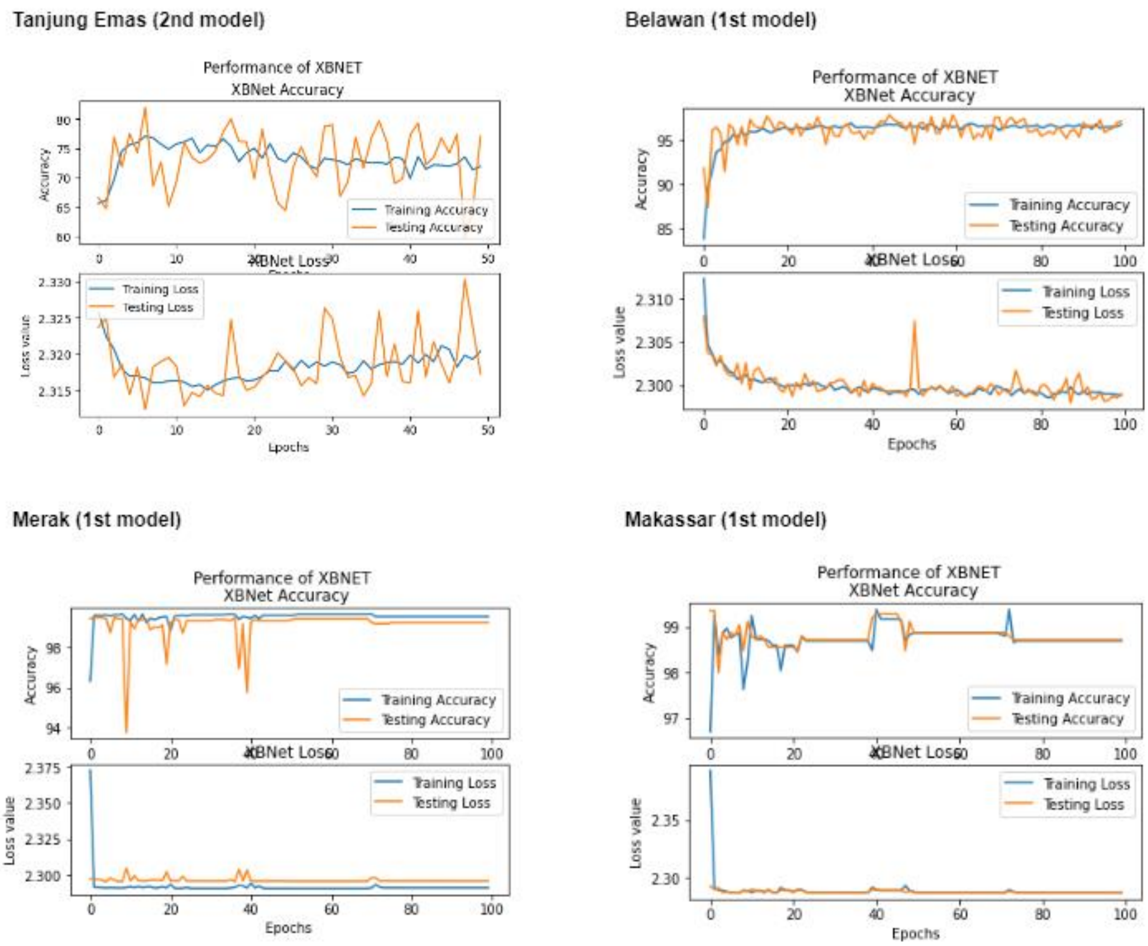


Figure 6 Comparison of Features Importance